# Computer-Vision-basierte Tracking- und Kalibrierungsverfahren für Augmented Reality

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

**DISSERTATION**

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

von
**Dipl.-Ing. Didier Stricker**
**aus Sarreguemines**

| | |
|---|---|
| Referenten der Arbeit: | Prof. Dr. José L. Encarnação<br>Prof. PhD. Gudrun Klinker |
| Tag der Einreichung: | 23.10.2002 |
| Tag der mündlichen Prüfung: | 29.11.2002 |

# Inhaltsverzeichnis

# Kapitel 1

# Einleitung

## 1.1 Motivation

Neben der real existierenden Welt ist in den letzten Jahren eine sogenannte Welt des Computers entstanden. Zwischen der künstlichen Computer-Welt und der realen Welt steht das Bindeglied der Mensch-Maschinen-Kommunikation. Diese ist trotz der rasanten Entwicklung der Computer-Technologie sehr eingeschränkt geblieben; die Kommunikation zwischen Mensch und Rechner findet in den meisten Fällen immer noch lediglich per Bildschirm, Tastatur und Mauskursor statt.

Im Folgenden wird gezeigt, dass *Augmented Reality* (AR) eine neue und mächtige Kommunikationsschnittstelle anbietet und somit viele neuartige Anwendungen und ein bessere Ausnutzung des Computer-Potentiales ermöglichen kann. Anschließend wird auf die technologische Realisierung eingegangen. Dabei stellen sich für AR die Kernprobleme des Benutzer-Trackings und der Registrierung virtueller Objekte in realen Umgebungen.

Um die Mensch-Rechner-Kommunikation zu verbessern, werden bislang im wesentlichen zwei Ansätze verfolgt: Einerseits wird versucht mit der Unterstützung von Spracherkennung, Bildverstehen, Gestik, künstlicher Intelligenz und Context-Awarness dem Rechner unsere Welt und unser Verhalten verständlich zu machen. Dieser Ansatz dient der Verbesserung der sogenannten "Rechner-Welt-Kommunikation". Andererseits soll auch die Kommunikation in der umgekehrten Richtung, also die "Welt-Rechner-Kommunikation", verbessert werden. Dazu wird dem Rechner beigebracht, sein Wissen (Daten + Algorithmen) in einer uns verständlicheren Form zu präsentieren. Die Sprachsynthese und Haptik (das Ausüben von Kräften) stellen beispielsweise Gebiete dieser Art der Kommunikationsoptimierung dar. Da der visuelle Sinn beim Menschen am dominantesten ausgebildet ist, eignet sich jedoch die graphische Datenverarbeitung als Übertragungsmedium besonders. Mit Hilfe der Visualisierung werden abstrakte Daten in Bilder umgewandelt, wodurch eine schnelle und intuitive Wahrnehmung der Informationen ermöglicht wird. Die *virtuelle Realität* (VR) stellt eines der Anwendungsbeispiele der grafischen Datenverarbeitung zur Optimierung der Welt-Rechner-Kommunikation dar. Mit Hilfe der VR-Technologie kann sich der Mensch in einer kompletten virtuellen Welt bewegen, mit ihr interagieren und sie dadurch sogar erleben.

Trotz der beiden Ansätze "Rechner-Welt-Kommunikation" und "Welt-Rechner-Kommunikation" bleiben die Rechner- und die reale Welt getrennt und stellen geschlossene Entitäten dar. Entweder befindet sich der Mensch in seiner realen Umgebung und kommuniziert

mit dem Rechner oder er taucht komplett in eine virtuelle Umgebung ein und lässt alle Bezüge zur realen Welt hinter sich. Um diese Trennung aufzuheben, muss das Problem gelöst werden, wie reale und künstliche Welt miteinander verknüpft werden können. Dies bedeutet, dass in einer realen Umgebung computergenerierte Informationen zugänglich gemacht werden müssen, indem virtuelle Objekte in die reale Welt eingegliedert werden und sich realen Objekten entsprechend verhalten. Die beschriebene Problemlösung wird durch die Technologie *Augmented Reality* (AR), einer virtuell erweiterten Realität der realen Welt, verwirklicht.

Damit die virtuellen Objekte in der realen Umgebung erscheinen, müssen sie bei der Umsetzung der Technologie AR direkt im Blickfeld des Benutzers eingeblendet werden. Diese Einblendung kann mit Hilfe zweier Methoden realisiert werden. Die erste Methode stellt einen sogenannten See-Through-Modus dar. Der Benutzer trägt eine Brille, in der die graphischen Objekte zusätzlich abgebildet werden. Diese eingeblendeten Informationen überlagern sich mit der realen Welt. Die zweite Methode wird Video-See-Through genannt. Hierbei wird eine Mini-Kamera auf der Brille des Benutzers montiert und die tatsächlichen Kamerabilder werden um virtuelle Objekte erweitert eingeblendet. In beiden Fällen kann eine korrekte Überlagerung nur erreicht werden, wenn für jedes gezeichnete Bild die Position, die Blickrichtung und der Blickwinkel der Kamera bzw. des Benutzerauges genau bekannt sind. Dies fordert sowohl eine sehr hohe Genauigkeit bei der Kalibrierung des Setups als auch bei dem Tracking der Benutzerbewegungen. Kleine Fehler können zu erheblichen Missregistrierungen der virtuellen Objekte oder zu sogenannten "jitter"-Effekten [1] führen. Die Kalibrierungs- und Tracking-Technologien stellen deswegen für AR eine Schlüsseltechnologie dar. Solange sie keine einwandfreie Positionsregistrierung ermöglichen, kann AR keinen realistischen Eindruck erzeugen. Die Zielsetzung besteht daher darin, sowohl Tracking-Geräte als auch Kalibrierungs- und Tracking-Technologien für AR zu optimieren.

Die üblichen Tracking-Geräte (z.B. magnetische oder mechanische Tracker), die beispielsweise für die Technologie *Virtuelle Realität* (VR) verwendet werden, liefern leider die für die Technologie AR erforderliche Genauigkeit nicht. Darüber hinaus benötigen sie immer eine Station (Empfänger) im Raum und haben meistens einen geringen Aktionsradius.

Erfolgsversprechende Ansätze stellen die optischen Methoden "Photogrammetrie" und "Computer-Vision" auf Basis üblicher Videokameras dar. Der Einsatz von AR erfordert, um virtuelle Objekte wirklichkeitsgetreu in die reale Welt einblenden zu können, die Kenntnis der genauen Kameraposition und -orientierung als auch ihrer Bewegungen. Durch die Detektion von Markern oder Szenenmerkmalen können die Bewegungen der Kamera zurückgerechnet und die Kamera hiermit als präzises Trackinggerät verwendet werden. Wesentlich für die Umsetzung von AR sind die verwendeten numerischen Verfahren, die sowohl ausreichende Genauigkeit und Stabilität als auch Echtzeitübertragung gewährleisten müssen. Für weitere Problemstellungen, wie beispielsweise die Berechnung des Benutzerblickwinkels, die Platzierung eines virtuellen Objektes in der realen Szene oder die 3D-Rekonstruktion für die Verdeckungsbehandlung, sind ebenfalls Computer-Vision-Techniken erforderlich. Dies heißt letztendlich, dass für AR die Brücke zwischen Computer-Vision und Computer-Graphik hergestellt werden muss, um Lösungen für die Kalibrierung- und Tracking-Problematik finden zu können.

[1] Zitterbewegungen des virtuellen Objektes

## 1.2 Problemstellung und Zielsetzung

Das Ziel dieser Arbeit ist, Verfahren und Lösungen zur korrekten Überlagerung von graphischen Objekten mit Bildern der realen Welt zu entwickeln.
Zwei grundsätzliche Problemstellungen werden hierbei analysiert. Die erste betrifft die Erfassung der geometrischen Parameter eines gegebenen Bildes, d.h. die Bestimmung des Blickpunkts, der Blickrichtung und des Blickwinkels der Kamera. Dieser Schritt wird als (Bild-)Kalibrierung bezeichnet und stellt eine wesentliche Technologie dar, da nur mit Hilfe einer exakten Kalibrierung ein virtuelles Objekt lagerichtig in einem Bild eingeblendet werden kann. Diese Thematik wird im Bereich der Computer-Vision intensiv erforscht. Nach einer ausführlichen Betrachtung und Beantwortung der Fragestellung, wie und unter welcher Voraussetzung Techniken der "Structure and Motion", "Pose estimation" oder "Bundle Adjustement" für AR einsetzbar sind, werden neue Ansätze zu den spezifischen AR-Problemen vorgestellt.
Hierbei werden sukzessiv Problemlösungen für einzelne Bilder, Bildfolgen und Live-Video berücksichtigt. Durch den Ansatz, die Kamera als ein präzises Trackinggerät einzusetzen, werden im Rahmen dieser Arbeit Lösungen für das bis jetzt ungelöste Problem des Echtzeit-Trackings erarbeitet. Die Verfahren sollen für mobile Rechner mit Standard-Hardware und mit und ohne Markerunterstützung anwendbar sein.
Die zweite Problemstellung betrifft die Gewinnung der 3D-Struktur der Szene aus 2D-Ansichten. Für eine realistische Wahrnehmung der betrachteten Szene muss die Konsistenz der 3D-Struktur gewährleistet sein, d.h. einerseits muss die Positionierung der virtuellen Objekte in der realen Welt ermöglicht und anderseits Verdeckungen zwischen realen und virtuellen Objekten behandelt werden. Um ein 3D-Modell für ein wahrnehmungsgetreues Rendering zu erzeugen, werden interaktive Lösungen zur 3D-Szenerekonstruktion auf Basis von 2 bis $N$ Bildern entwickelt.

## 1.3 Organisation der Arbeit

Die vorliegende Arbeit ist in fünf Kapitel untergliedert.
In der Einführung wird zunächst die Technologie AR vorgestellt und auf die Kalibrierungs- und Tracking-Problematik in Augmented-Reality eingegangen. Der anschließende Abschnitt legt die Problemstellung und Zielsetzung der Dissertation fest. Die wichtigsten Ergebnisse werden in zusammenfassender Form genannt.
Im zweiten Kapitel (Kapitel 2) wird die Technologie *Augmented-Reality* und dafür wesentliche Begriffe erläutert. Einen besonderen Schwerpunkt besitzt in diesem Kapitel die Diskussion der AR-Trackingproblematik.
Das Kapitel 3 zeigt sowohl Grundlagen als auch neue, für das Verständnis dieser Arbeit wesentliche, theoretische Ergebnisse der Computer- Vision auf.
Zur Behandlung der Problematik von AR wird eine grundsätzliche Trennung zwischen Off-line- und Echtzeit-Anwendungen getroffen. Dafür werden in dem Kapitel 4 die Begriffe "Augmented Image" und "Augmented Video" eingeführt. Das Kapitel 4 gliedert sich in drei Abschnitte. Einleitend wird die Problematik der Erweiterung eines einzelnen Bildes bearbeitet und zahlreiche Algorithmen zur Berechnung der Kameraposition und -orientierung anhand von 3D-Punkten implementiert, verglichen und evaluiert. Anschließend wird der Fall mehrerer Sichten einer Szene untersucht und flexiblere Lösungen zur

Erweiterung des Bildes mit graphischen Objekten entwickelt. Das Kapitel schließt mit der Betrachtung der Bearbeitung ganzer Bildfolgen. In diesem Zusammenhang wird ein Automatisierungsmechanismus präsentiert, der mit Hilfe von synthetischen sowie realen Bildern evaluiert wird.
Die Kapitel 5 und 6 setzen sich mit optischem Echtzeit-Tracking auseinander. Im Kapitel 5 wird ein neues markerunterstütztes Trackingverfahren von der Bildverarbeitung bis zur Bestimmung der Kamerabewegung im Detail vorgestellt. Ein Rahmensystem (VBT) für ein optisches Tracking wird entwickelt und zwei Lösungsverfahren VBT-I und VBT-II mit jeweils schwarz-weißen und farbigen Markern eingehend betrachtet.
Eine neuentwickelte, markerlose Tracking-Methode, die auf dem Prinzip der Bildregistrierung basiert, wird im folgenden Kapitel 6 vorgestellt und hinsichtlich zwei weiterer Verfahren diskutiert.
Die Vielfältigkeit der AR-Anwendungsmöglichkeiten wird anhand einzelner Beispiele in den Bereichen Bauwesen, Produktion und Service, persönliche AR-Informationssysteme für archäologische Stätten, Computer-Spiele und Kunst in den Kapiteln 4, 5 und 6 konkret verdeutlicht.
Abschließend werden die wichtigsten Ergebnisse im Kapitel 7 zusammengefasst und ein Ausblick auf zukünftige Forschungsgebiete und Entwicklungsmöglichkeiten gegeben.

## 1.4 Zusammenfassung der wichtigsten Ergebnisse

Im Rahmen dieser Arbeit wurden Computer-Vision Methoden zur Lösung der Augmented-Reality Kalibrierungs- und Trackingprobleme entwickelt. Insbesondere wurden (1) neben der “structure and motion” Methode, die theoretischen Grundlagen eines neuen Verfahrens (“Calibration propagation” [26]) als auch ihre praktische Umsetzung erarbeitet, (2) optische Lösungen zum Echtzeit-Tracking mit Hilfe von schwarz-weißen sowie farbigen Markern ermöglicht und (3) ein neues Verfahren zum markerlosen Tracking konzipiert und implementiert.
Um eine klare Abgrenzung zur Augmented-Reality zu ermöglichen, wurden in dieser Arbeit zunächst die Begriffe “Augmented-Image” und “Augmented-Video” eingeführt. Nach einer detaillierten Untersuchung der Off-Line-Erweiterung wurden robuste Verfahren zur Bearbeitung eines einzigen Bildes als auch mehrerer Bilder entwickelt und implementiert. Insbesondere wurden fünf verschiedene Algorithmen zur Berechnung der Kameraposition und -orientierung mit vier, fünf und sechs Punkten implementiert und auf Basis zahreicher Simulationen verglichen. In dem Fall, dass mehrere Ansichten zur Verfügung stehen, wurden “structure and motion” Algorithmen für planare und nicht-planare Szenen analysiert und implementiert. Darüber hinaus erfolgte sowohl die Entwicklung der theoretischen Grundlage als auch die praktische Umsetzung des neuen, sogenannten “Calibration Propagation”-Verfahrens [26]. Das Verfahren besitzt die Zielsetzung, ein Bild, das von einer unbekannten Kamera aufgenommen wurde, an Hand eines zweiten Bildes, für welches alle Kameraparameter bekannt sind, zu kalibrieren und mit virtuellen Informationen zu erweitern. Dieses Verfahren wurde “Calibration Propagation” genannt, da die Kalibrierungsinformationen von einem kalibrierten auf ein unkalibriertes Bild übertragen werden. Mit dem Verfahren können zeitgleich sowohl Zoomänderungen als auch Kamerabewegungen erfasst werden.
Bezüglich der automatischen Kamerabewegungsrekonstruktion wurde für Bildfolgen ein

globales Konzept erstellt, welches im ersten Schritt starke Bewegungen kompensiert und anschließend durch Subpixel-Korrelationsmethoden Punkte verfolgt. Verfeinerungen der Bestimmung der 3D-Kamerabewegung wurden mit globalen Optimierungsalgorithmen, sogenannte "Bundle Adjustements", auf Basis von M-Estimatoren erzielt, die sogenannte "outliers" zulassen.
Des weiteren erfolgte die Problembearbeitung der Behandlung von Verdeckungen zwischen realen und virtuellen Objekten. Die Lösung des Problems erfordert für ein wahrnehmungsgetreues Rendering der Szene ein 3D-Modell. Auch in diesem Bereich konnte gezeigt werden, dass die benötigten Informationen allein aus den Bildern gewonnen werden können [39]. Die 3D-Modelle werden hierbei interaktiv aus zwei oder mehr Bildern erzeugt, wobei eine Fehleranalyse den Benutzer über die Genauigkeit der Rekonstruktion informiert. Alle Verfahren wurden in ein System integriert, das eine flexible Vorgehensweise zur Erweiterung von Bildern und Bildfolgen mit minimalem Arbeits- und Zeitaufwand ermöglicht.
Im Bereich des optischen Echtzeit-Trackings mit Markern wurde ein neues Verfahren, das auf einem "Punkt-zur-Linie"-Distanzminimierungsverfahren basiert, entwickelt [23]. Das daraus resultierende Trackingsystem erfüllt die AR-Anforderungen bezüglich Genauigkeit und Echtzeit-Performance und ermöglicht die Demonstration einer AR-Anwendung mit See-Through Head Mounted Displays, wie zum Beispiel der Montage eines Autotürschlosses [30] oder Einblendung von Bauplänen im Bauwesen [42]. Das neue VBT-System stellt ein modulares System dar, das in zwei Versionen existiert. Diese basieren sowohl auf der Verwendung von schwarz-weißen als auch farbigen Markern und können mit einem Laptop mit herkömmlichen Kameras betrieben werden. Der sogenannte ArBrowser, ein AR-System in einem Internet-Browser, basiert auf der VBT-Trackingkomponente.
Einen weiteren wichtigen Beitrag stellt die Entwicklung eines markerlosen Trackingverfahrens für mobile AR-Systeme dar [28, 22, 27, 24, 89, 25]. Hierfür wurde der Begriff der Stützung für optische Tracker eingeführt und definiert. Die Stützung, die beispielsweise von Markern gegeben werden kann, wurde in Form von Kamerabildern zur Verfügung gestellt. Das Tracking-System basiert auf einer Reihe von vordefinierten Bildern, auch Referenzbilder genannt, mit denen das Live-Videobild verglichen wird. Bei Existenz einer Zuordnungsmöglichkeit wird die Transformation vom Referenzbild zum Live-Video-Bild mit Hilfe von Bildregistrierungsverfahren berechnet. Diese Bildregistrierungsverfahren wurden intensiv untersucht, wobei Ansätze auf Intensitäts- und Fourierbasis ausgewählt und implementiert wurden. Das Tracking konnte mit dem Fourier- Ansatz erfolgreich umgesetzt und sowohl Indoor als auch Outdoor demonstriert werden.

# Kapitel 2

# Augmented Reality

In diesem Kapitel wird zuerst Augmented Reality (AR) definiert und im globalen Kontext von Mixed Reality (MR) vorgestellt. Dann werden die notwendigen Technologien, die Bestandteile von AR-Systemen sind, erläutert. Insbesondere wird die Tracking-Problematik und dazugehörige wichtige Begriffe erläutert. Eine Übersicht möglicher Anwendungsgebiete verdeutlicht abschließend das starke Potential dieser Technologie.

## 2.1 Definitionen

### 2.1.1 Augmented Reality

Augmented Reality (AR) wird als eine Technologie, die virtuelle Objekte mit der realen Welt kombiniert, definiert. Die Innovation von AR besteht darin, dass graphische Objekte direkt im Benutzersichtfeld bzw. in einem Live-Kamerabild eingeblendet werden. Dadurch werden sie wie reale Objekte der Szene wahrgenommen.
Der Begriff *Augmented Reality*, der für *erweiterte Realität* steht, wurde von David Mizell und Thomas Claudell Anfang der 90er Jahre geprägt [17]. Auf der Suche nach neuen Möglichkeiten, Flugzeugmechaniker bei Montage- und Wartungsaufgaben zu unterstützen, entwickelten Mizell und Claudell diese Technologie. Ziel war, komplizierte Montagepläne durch direkt am realen Objekt eingeblendete Hinweise zu ersetzen und dadurch unnötige Zeitverluste zu vermeiden [64, 19]. Die weitere Entwicklung von AR erfolgte jedoch hauptsächlich in Forschungs- und Universitätsbereichen (siehe beispielsweise [33, 81, 74, 76] ).

### 2.1.2 Charakterisierung von Augmented Reality

AR stellt ein neues Gebiet im Bereich der Computertechnologien dar, das sicherlich noch eine starke Entwicklung und auch Neudefinition erfahren wird, das aber schon mit den folgenden Eigenschaften zu charakterisieren ist [7]:

1. **AR kombiniert reale und virtuelle Welt**

   Der Benutzer befindet sich in einer realen Umgebung, in der weitere, virtuelle Informationen zur Verfügung gestellt werden. Der neue Aspekt von AR besteht darin, dass die Computerdaten direkt in das Benutzerumfeld eingebracht werden. Diese Lösung steht im Gegensatz zu dem Ansatz von VR, bei dem sich der Benutzer in

einer kompletten synthetischen Welt, die durch immersive Lösungen erzeugt wird, befindet.

Die beiden folgenden Abbildungen zeigen zwei Anwendungsbeispiele von AR. Abbildung (a) verdeutlicht, wie zu Anschauungszwecken eine virtuelle Brücke in eine reale Umgebung eingeblendet werden kann. Abbildung (b) zeigt die Nutzung von AR zur Unterstützung von Montagearbeiten.

(a) (b)

Abbildung 2.1: Einblenden einer virtuellen Brücke in eine reale Umgebung (a) und Beispiel einer Montageunterstützung mit AR (b)

2. **AR ist interaktiv in Echtzeit**

   Als erste Interaktionsanforderung wird die Bedingung gestellt, dass der Benutzer sich frei bewegen kann. Als Mindestanforderung müssen Kopfbewegungen, die eine Veränderung seines Blickpunktes auf die Szene bewirken, möglich sein. Das System muss hierbei in der Lage sein, die AR-Szene mit Hilfe von Trackingsystemen konsistent zu halten.

   Neben den Interaktionen mit virtuellen Objekten, die aus VR bekannt sind, treten auch Wechselwirkungen mit realen Szenekomponenten auf. Wenn beispielsweise ein reales Objekt bewegt wird, kann es ein virtuelles Objekt verdecken. Die Geometrie und die neue Position des bewegten Objektes muss bekannt sein, um eine korrekte Darstellung der AR-Szene zu ermöglichen. Dies bedeutet, dass alle verursachten Szeneänderungen vom System berücksichtigt werden müssen, was im idealen Fall eine kontinuierliche Analyse der realen Szene erfordert [42].

3. **AR setzt drei-dimensionale Registrierung voraus**

   Das virtuelle Objekt muss sich der Szene anpassen, d.h. es muss in 3D exakt positioniert werden. Das Einblenden von Texten, Videos oder Photomontagen wird nicht als AR bezeichnet, da diese Informationen nur 2D-Darstellungen und nicht in die Szene integriert sind.

AR bietet eine neue Form der Datenpräsentation und der Rechnerkommunikation. Auf Grund der Nutzung des visuellen Sinnes ermöglicht AR eine wesentlich effizientere Kommunikation zwischen Mensch und Rechner als traditionelle Interface-Geräte. Deswegen wird AR auch als eine neue *Menschmaschine-Schnittstelle* betrachtet [42].

### 2.1.3 Mixed Reality

Die Bezeichnung *Mixed Reality* (MR) wurde von Paul Milgramm im 1994 eingeführt, um alle möglichen Kombinationsformen von realen und künstlichen Umgebungen zu erfassen und einzuordnen [63]. MR bildet ein Kontinuum, das von der realen und virtuellen Welt begrenzt ist und entlang dessen der Anteil der künstlichen Daten variiert (siehe Abbildung 2.2). Daraus folgt, dass die reale Benutzer- und synthetische Computerwelt nicht mehr als zwei getrennte Welten betrachtet werden, sondern es existieren alle möglichen gemischten Varianten der zwei Welten, die sie miteinander verbinden.

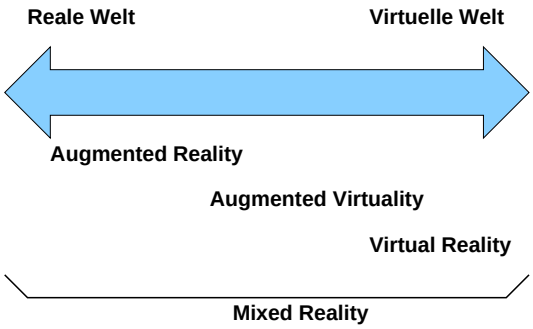



Abbildung 2.2: Mixed Reality Kontinuum (Paul Milgramm [63])

Die Erweiterung einer realen Umgebung mit neuen virtuellen Objekten wird als *Augmented Reality* (AR) bezeichnet. Umgekehrt wird das Einblenden von realen Szeneausschnitten in einer virtuellen Welt als *Augmented Virtuality* (AV) definiert. Diese letzte Technologie ist schon längere Zeit bekannt und wird z.B. in virtuellen Studios verwendet.

### 2.1.4 Off-line Bearbeitung von Bildern und Videos

Für einige Anwendungen, wie z.B. der Präsentation eines neuen CAD-Modells in seiner realen Umgebung, werden nur wenige Ansichten benötigt (siehe Abbildung 2.1 "Virtuelle Brücke" im Abschnitt 2.1.2). In diesen Fällen wird von *Augmented Images* gesprochen. Auf die selbe Weise führt die Off-line Bearbeitung von Videosequenzen zu sogenannten *Augmented Videos*. Diese beiden Gebiete werden im Kapitel 4 thematisiert.

## 2.2 AR-Systeme

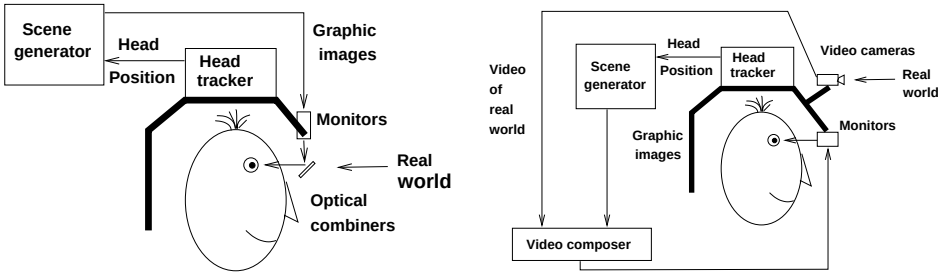
In vorliegendem Abschnitt werden die Hauptkomponenten eines AR-Systems zusammengefasst. Einleitend wird die Präsentationstechnologie erläutert und verschiedenen Alternativen vorgestellt. Anschließend wird die Problematik der Benutzerlokalisierung vorgestellt und die Anforderungen an die Interaktionsgeräte und die Rechnereinheit behandelt.

### 2.2.1 Präsentation

Um den Eindruck von erweiterter Realität zu vermitteln, müssen drei-dimensionale Objekte in ein reales Bild integriert werden. In erster Linie existieren drei Darstellungsmöglichkeiten, die im folgenden vorgestellt werden.

### Head Mounted Display

Im idealen Fall wird das virtuelle Objekt, wie in Abbildung 2.3(a) vorgestellt, direkt im Benutzersichtfeld in Stereo eingeblendet. Diese Technik ist mit Hilfe einer durchsichtigen Brille oder eines Spiegels, in den nur die für das linke bzw. rechte Auge entsprechende Graphik gezeichnet wird, möglich. Das neue Objekt erscheint dadurch in 3D mit einer gegebenen Tiefe in der realen Szene. Displays dieser Art werden *See-Through Head Mounted Display* (See-Through HMD) genannt. Die Lichtverhältnisse zwischen realer Welt und überlagerten Objekten stellen hier einen entscheidenden Faktoren für die Qualität der Wahrnehmung der AR-Szene dar. Grundsächlich leiden See-Through Lösungen mit den heutigen HMD's unter Helligkeitsproblemen: durchsichtige Displays sind sehr dunkel, so dass ohne optimale Beleuchtung oder bei dunklen Szeneteilen eine praktische Anwendung schnell unmöglich wird.




See-Through Display (a) Video-See-Through Display (b)

Abbildung 2.3: Schematische Darstellung beider Präsentationskonzepte

Eine Alternative besteht in der Verwendung des Video-Modus, auch *Video-See-Through* genannt. Hierbei werden Bilder von einer oder zwei Fingerkameras, die am HMD montiert sind, eingeblendet (Abb. 2.3 (b)). Diese werden in Echtzeit bearbeitet und mit den synthetischen Daten vervollständigt. Bei Videodarstellungen kann auf die Möglichkeit, die Bilder durch numerische Algorithmen zu verbessern, (Histogrammausgleich, Kontrastverstärkung, usw.) zurückgegriffen werden.
Im Video-Modus können auch Szenendetails eingezoomt werden, was in Anwendungsgebieten wie zum Beispiel der Medizin von großem Nutzen ist. Nachteilig wirkt sich jedoch aus, dass die Umgebung trotz hoher Darstellungsqualität nicht mehr direkt und natürlich wahrgenommen wird.

### Monitor

Ein AR-System kann auch auf Monitorbasis entwickelt werden. Eine Kamera nimmt die Szene auf, die dann auf dem Monitor zu sehen ist. Um eine 3D-Ansicht zu erzeugen, werden hierbei Stereo-Kameras und Stereo-Brille eingesetzt.

(a) (b) (c)

Abbildung 2.4: AR mit Monitoren: Hand-held display (a), PC-Monitor (b) und Projektionsleinwand (c)

Dieser Ansatz ist je nach Applikation ausgehend vom Hand-Held Monitor (Abbildung 2.4(a)), der über Szenenteile bewegt wird, über einen Standard PC-Monitor (Abbildung 2.4(b)), bis auf Größe einer Projektionsleinwand skalierbar (Abbildung 2.4(c)). Wie für HMD sind auch durchsichtige Displays denkbar, wobei der Monitor und die Position des Benutzerkopfes registriert sein müssen.

### Projektor

Mit Hilfe von Projektoren können die virtuellen Informationen direkt auf Objekte der realen Szene projiziert werden [9]. Der wesentliche Vorteil ist, dass der Benutzer kein Display oder HMD tragen muss und die Generierung der erweiterten Szene unabhängig von seinem Blickpunkt auf die Szene ist. Das bedeutet auch, dass mehrere Benutzer ohne weiteren Aufwand die selbe Szene betrachten können.
Dennoch müssen folgende Nachteile dieser Technologie beachtet werden:

- Die komplette Geometrie der realen Objekten muss bekannt sein,
- jede Änderung der realen Szene muss erfasst werden,
- eine gute Darstellungsqualität kann nur in abgedunkelten Räumen für Objekte einfacher Form erreicht werden
- und letztendlich ein reales Objekt, auf das projiziert wird, muss immer vorhanden sein; d.h. es kann kein zusätzliches, völlig virtuelles Objekt in die Szene eingefügt werden.

## 2.2.2 Benutzerlokalisierung

Die Benutzerlokalisierung, auch *"Tracking"* genannt, stellt zur Zeit die größte technische Herausforderung bei der AR-Anwendung dar. Aus jedem beliebigen Blickpunkt des Benutzers müssen die virtuellen Objekte lagerichtig in der realen Welt erscheinen. Daraus folgt, dass :

- die Geometrie der realen Szene,
- die Position, Orientierung und Skalierung der virtuellen Objekte in der realen Szene
- die Position, Orientierung und Blickwinkel des Benutzers

zu jeder Zeit bekannt werden müssen.

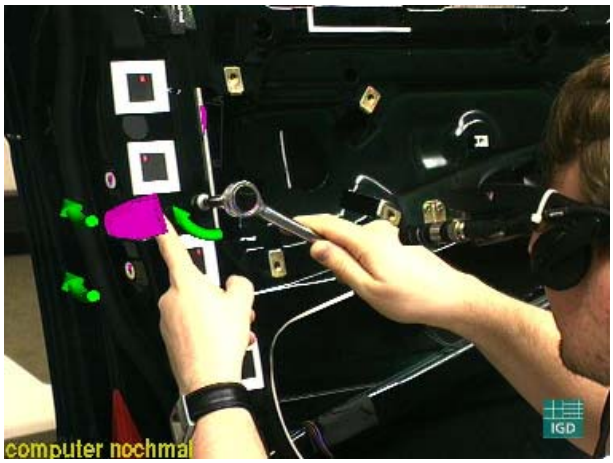



Abbildung 2.5: Augmented Reality Anwendung mit optischem Tracking

Die Trackingtechnologien, die für die virtuelle Realität (VR) angewendet werden, sind für AR leider nicht ausreichend genau. Eine VR-Umgebung stellt eine abgeschlossene Welt dar. Kleine Ungenauigkeiten üben so gut wie keine Fehler aus, da sie kaum wahrnehmbar sind. Die VR-Szene bleibt immer in sich konsistent. Auf Grund des direkten Bezuges zur realen Szene führen jedoch kleine Abweichungen bei AR zur fehlerhaften Registrierung zwischen virtuellen und realen Objekten, wodurch der gesamte Eindruck beeinträchtigt wird.
Deswegen wurden in den letzten Jahren neue optische Tracker entwickelt. Für die Bildregistrierung werden Marker in der Szene platziert, die in den Videobildern der Kamera extrahiert und verfolgt werden. Da die Position der Marker in 3D bekannt ist, kann die Kameraposition aus den Bildern zurückgerechnet werden. Mit Hilfe dieses Verfahrens, das für die erste Prototypisierung eines AR-Systems verwendet wurde [23], kann bereits eine gute Überlagerungsgenauigkeit erzielt werden.
Dennoch ist auch dieses Verfahren begrenzt, da die Marker immer sichtbar sein müssen, und es nicht immer möglich ist, sie überall in die Szene einzubringen. Des weiteren ist die Messrate ohne spezielle Hardware noch auf maximal 25 Hz begrenzt.
Ein weitere Verbesserung besteht in der Verwendung des Hybrideansatzes, bei dem die Geräte so ausgewählt werden, dass sich ihre Stärken und Schwächen gegenseitig ausgleichen lassen. Die verschiedenen Sensordaten liefern durch Fusion oder hierarchische Bearbeitung eine sichere und präzise Lösung. So können beispielsweise Trägheitssensoren schnelle Bewegungen erfassen und Kameras die gewünschte Präzision ermitteln.

### 2.2.3 Interaktionen

AR verlangt neue Interaktionsmethoden, die eine höhere Flexibilität und ein breiteres Anwendungsgebiet als eine herkömmliche Tastatur-Maus-Kombination ermöglichen. Zu

diesen neuen Methoden zählen Spracheingabe, Gesteneingabe, magnetische/optische Pointer und Geräte wie Unterarm- oder Handtastaturen bis zu Sensoren zur Verfolgung von Szenenobjekten.

### 2.2.4 Wearable Computer

Der Benutzer muss an der Stelle, wo die Aufgabe zu erfüllen ist, Zugang zu den benötigten Informationen haben.

Abbildung 2.6: AR mit Wearable Computer

Aus diesem Grund müssen die Rechner tragbar und leicht sein. Netzverbindungen, um neue Daten holen oder schicken zu können, müssen hierbei berücksichtigt werden.

## 2.3 Die Tracking-Problematik

### 2.3.1 Begriffsdefinition

#### Statische Fehler

Statische Fehler treten in erster Linie infolge eines Präzisionsmangels der Trackinggeräte auf. Dazu addieren sich systematische Berechnungsfehler, die z.B. durch falsche Evaluierung von Transformationen zwischen den verschiedenen Systemkomponenten, wie beispielsweise Trackingsensoren und Displays, verursachen werden können. Des weiteren müssen auch Fehler der Darstellungsparameter, wie z.B. die Wahl einer falschen Brennweite, aspect ratio usw., berücksichtigt werden.
Wenn die statischen Fehler klein und mit Rauschen zu assimilieren sind, erscheint das virtuelle Objekt richtig registriert aber in der zeitlichen Abfolge kann immer noch ein Fehler auf Grund der Messdatenschwankungen wahrnehmbar sein. Das Objekt scheint leicht zu “zittern”. Dieses Phänomen wird als “jitter”-Effekt bezeichnet und kann z.B. mit Hilfe von Filtern geglättet werden.

#### Dynamische Fehler

Dynamische Fehler resultieren aus Latenzen, die durch zu geringe Messrate, Bearbeitungs- und Darstellungszeit erscheinen. Solche Fehler besitzen besondere Wirkungen im See-Through-Modus, da das virtuelle Objekt immer mit einem Versatz gegenüber der realen Umgebung erscheint.

Größenordnungen für die statischen und dynamischen Fehler sind in der Tabelle 2.1 dargestellt.

| Fehlerart | Größenordnung |
|---|---|
| Position | wenige Millimeter |
| Winkel | 0.1 Grad |
| Durchsatz | 20 Hz |
| Latenz | 2ms |

Tabelle 2.1: Fehlerwerte

**Outside-In-Systeme**

Für ein Outside-In-Trackingsystem stellen die Sensoren, die im Raum angebracht werden, eine feste Station dar. Die Tracker-Referenzen oder Marker werden an das zu verfolgende Objekt befestigt.

**Inside-Out-Systeme**

Für ein Inside-out-Trackingsystem werden die Sensoren an dem Objekt befestigt und die Referenzen werden im Raum angebracht.

**Inside-Out versus Outside-In**

Im Vergleich zu den Outside-In-Sytemen besitzen die Inside-Out-Systeme den Vorteil einer genaueren Orientierungsbestimmung bei Anwendung derselben Technologie [10].
Dies wird an Hand eines einfachen Beispiels veranschaulicht:
Für eine typische HMD-Anwendung beträgt der Abstand zwischen Zielobjekt, dem HMD, und der Rotationsachse, dem Hals, ca. 20 cm. Zwischen dem HMD, an dem die Marker oder Sensoren befestigt sind und den Objekten in der Szene beträgt der Abstand ungefähr 2 Meter. Eine Rotation des HMD von 0.1 Grad wird angenommen.

- Für das Outside-In System werden die Marker am HMD erfasst. Eine reine Drehung von 0.1 Grad entspricht einer räumlichen Verschiebung von ca. $20 \times tan(\pi/180 \times 0.1)$, d.h. 0.035 cm. Das System muss deshalb in der Lage sein, Bewegungen von mindestens 3.5 mm bei einer Entfernung von 2 m zu erfassen.

- Für das Inside-Out System wird der Sensor am HMD montiert und die Referenzen werden in der Szene in 2 m Entfernung angebracht. Eine Rotation von 0.1 Grad bewirkt eine Verschiebung von $200 \times tan(\pi/180 \times 0.1) = 0.35$ cm. Die Sensoren müssen in diesem Fall nur Bewegungen von 0.35 cm erfassen können.

Das Beispiel verdeutlicht, dass der Outside-In-Tracker für dieselbe Anwendung eine, 10 mal bessere Auflösung und Genauigkeit als der Outside-In-Tracker liefern muss, um vergleichbare Anforderungen zu erfüllen.

### 2.3.2 Tracking-Anforderungen für AR

#### Hohe Genauigkeit

Die Genauigkeit wird durch die Größe des Positions- und Orientierungsfehlers des Trackers bestimmt. Ein wesentliches Problem in AR besteht darin, dass eine sehr hohe Genauigkeit erforderlich ist, um einen realistischen Eindruck zu erzeugen. Kleine Fehler üben eine starke Wirkung aus und sind sofort erkennbar. Insbesondere verursachen bereits geringe Abweichungen der Orientierungsbestimmung gravierende Inkonsistenzen. So bewirkt beispielsweise ein Fehler von 1.5 Grad für ein virtuelles Objekt, das 1 m vom Benutzer entfernt ist, eine Missregistrierung von 2.6 cm und schon 5,2 cm für ein 2 m entferntes Objekt.

#### Hohe Auflösung

Die Auflösung entspricht der kleinsten Positions- und Orientierungsänderung, die vom Tracker registriert wird. Wie bei der Genauigkeit ist eine hohe Auflösung wesentlich, um kleine Benutzerbewegungen erfassen zu können.

#### Hoher Durchsatz

Hier ist der Echtzeitanforderung Rechnung zu tragen. Um eine flüssige graphische Darstellung zu erhalten, sind mindestens 20 Bilder pro Sekunde nötig. Diese Geschwindigkeit sollte ein AR-System mindestens besitzen.

#### Niedrige Latenz

Um eine niedrige Latenz zu erzielen, muss nicht nur die Framerate sondern vor allem die Verzögerung betrachtet werden. Die Verzögerung verursacht vor allem bei der optischen Überlagerungen Probleme, da durch sie ein zeitlicher Versatz zwischen realer und virtueller Welt entsteht.
Im Video-Modus spielen Latenzfehler unter der Voraussetzung, dass die dargestellten Bilder mit dem Trackinggerät richtig synchronisiert sind, keine große Rolle. Die Latenz wirkt sich nur durch eine globale Verzögerung in der Wahrnehmung aus. Im Gegensatz zum Video-Modus verursachen kleine Verzögerungen bei See-Through Anwendungen falsche Registrierungen. Das Objekt wird in diesem Fall mit veralteten Positionsparametern in der Szene gerendert und dadurch falsch registriert. Daraus resultiert ein sogenannter “Schwimmeffekt” des virtuellen Objekts in der Szene. Höhere Messtakte und Prädiktion der Kopfsposition sind die zwei einzigen Möglichkeiten, diese Fehler zu reduzieren.

#### Verträglichkeit, Gewicht und Arbeitsvolumen:

Die Anforderungen an das Gewicht und an das Arbeitsvolumen des Tracking-Systems sind von der Anwendungen abhängig. Generell sollte das Gewicht am Kopf so niedrig wie möglich sein. Das angestrebte Gewicht eines HMD in Kombination mit einem Trackingsensor sollte nicht mehr als 100 bis 150 g betragen.
Die Verträglichkeit soll vor allem gegenüber magnetischen Feldern oder Metall gewährleistet sein. Die eingegebenen Genauigkeiten sollen im ganzen Arbeitsvolumen gelten.

## 2.4 Anwendungsgebiete

In den folgenden Abschnitten wird ein Einblick in die unterschiedlichen Anwendungsgebiete von AR gegeben. Eine umfassende Zusammenstellung zu den vielfaltigen, aktuellen Einsatzgebieten von AR kann in [75, 6] nachgelesen werden.

### 2.4.1 Architektur

Häufig werden Architektur-Modelle mit CAD-Tools entworfen und auf dem Monitor oder in Photomontage dem Kunden präsentiert. Mit AR können diese Modelle in verschiedene Bilder oder auch in ein Video eingeblendet werden. Diese Technik weist den großen Vorteil auf, dass das Modell in die reale Szene mit korrekter Perspektive integriert wird und realistisch erscheint.

Abbildung 2.7: Einblenden von CAD-Modellen in deren realen Umgebungen

Das Modell kann interaktiv mit dem tasächlichen Hintergrund editiert und nachjustiert oder durch ein anderes CAD-Modell ausgetauscht werden. Im Gegensatz zu VR muss nicht die ganze Szene nachmodelliert werden. Dadruch wird Zeit gespart und der optische Eindruck ist wesentlich realistischer.

### 2.4.2 Simulationsüberprüfung

Viele physikalischen Zusammenhänge werden weitestgehend per Computer simuliert. Oft ist es jedoch nur sehr schwer möglich, die Ergebnisse mit der Realität gegenüberzustellen. Eine visuelle Überlagerung bietet eine schnelle globale Überprüfung des Zusammenhanges an. Diese Möglichkeit der Datenkontrolle kann beispielsweise bei der Auswertung von Crash-Tests eingesetzt werden. Durch die Unterstützung von AR kann leicht überprüft werden, ob sich die Fahrzeugkarroserie entsprechend der berechneten Simulation verformt.

### 2.4.3 Medizin

In der Medizin gibt es grundsätzlich das Problem, dass Sensordaten beispielsweise bei einer Operation über einen Monitor gelesen werden müssen. Idealerweise könnten sie jedoch mit Hilfe von AR direkt in das Sichtfeld des behandelnden Arzt eingeblendet werden. Der Artz würde die Informationen schneller und ohne Ablenkung erhalten und könnte sie somit wesentlich leichter auswerten. An ein medizinisches system werden selbstverständlich sehr hohe Präzisions- und Zuverlässigkeitsanforderungen gestellt.

### 2.4.4 Montage und Service

Da Maschinen immer schneller und komplexer entwickelt werden, besteht besonders im Montage- und Service-Bereich ein erhöhter Bedarf, den Monteur bzw. Anwender bei seiner Arbeit zu unterstützen. Bisher stehen meistens nur Papierdokumente und manchmal Videos, die Arbeitsschritte einer Reparatur oder einer Montage zeigen, zur Verfügung. Als eine der ersten Firmen untersucht die Fluggesellschaft Boeing ein AR-System zur Montageunterstützung von Kabelverlegungsarbeiten (siehe Abbildung 2.8(a)).

(a) (b) (c)

Abbildung 2.8: AR-Unterstützung bei Montageaufgaben

Schritt für Schritt wird dem Arbeiter gezeigt, wie und wo die Kabel zu verlegen sind. Eine ähnliche Anwendung zeigt durch Einblenden einer graphischen Animation mit Textausgabe, wie ein Schloss in eine Fahrzeug-Tür einzubauen oder wie eine Maschine zu reparieren ist (siehe Abbildung 2.8(b,c)).

### 2.4.5 Persönliche Informationssysteme

Geräte, wie z.B. Mobiltelefone oder PALM-Rechner, weisen zunehmend Rechnerleistungen auf und besitzen zum Teil bereits integrierte Mini-Kameras oder einen USB- oder IEEE-1934 Port.
Durch einfache Verfahren ist es möglich Bilder mit Text oder Graphik zu überlagern. Auf Grund der geringen Größe der genannten Geräte bietet sich somit die Möglichkeit, mit Hilfe von AR-Anwendungen beispielsweise Stadt- oder Museumsführungen zu realisieren. Der Anwender/Tourist kann dabei in einer Stadt oder auf einem archäologischen Gelände geführt und gleichzeitig informiert werden.

Abbildung 2.9: Virtueller Tempel in seinem realen Kontext

Wie in Abbildung 2.9 gezeigt, können Modelle von zerstörten Tempeln in ihrem realen Kontext eingeblendet und somit Wissen über Kulturerbe besser vermittelt werden.

## 2.5 Zusammenfassung

In diesem Kapitel wurde die Technologie Augmented-Reality und die dazugehörigen Systemkomponenten vorgestellt. Insbesondere wurde die Anforderungen für das Tracking abgeleitet und auf den aktuellen Lösungsmangel und Forschungsbedarf hingewiesen. Zum Schluss wurde anhand von Anwendungsbeispielen die Vielfalt der möglichen AR-Einsatzgebiete verdeutlicht. Die Bearbeitung von einzelnen Bildern und Videos wird bereits heute, z.B. in der Film-Industrie, in die Praxis umgesetzt, jedoch fehlen flexible und schnelle Lösungen für Echtzeit-Anwendungen, um eine weitere Verbreitung zu ermöglichen.

# Kapitel 3

# Computer-Vision

Die Bestimmung der Kameraparameter spielt eine zentrale Rolle in dem Erweiterungsprozess eines Bildes. Jedoch liegen oft sehr wenig Informationen über die Kamera und die 3D-Geometrie der Szene vor. Die fehlenden Parameter und Daten müssen mit Hilfe der verschiedenen Szeneansichten auf die Szene zurückgewonnen werden. Mit dieser Problematik beschäftigt sich der Bereich "Computer-Vision", der Thema dieses Kapitels ist.
Die grundlegenden Definitionen und wesentlichen Algorithmen der Computer-Vision, die auch die Basis für AR bilden, werden hierbei vorgestellt. Im ersten Abschnitt wird das Kameramodell beschrieben und die Grundprinzipien der Kamerakalibrierung erläutert. Anschließend wird auf die Geometrie und die Zusammenhänge mehrerer Bilder näher eingegangen. Insbesondere werden hierbei die Methoden zur Bestimmung der Kamerabewegung aus mehreren Bildern und zur Szenerekonstruktion untersucht.

## 3.1 Das Kameramodell

### 3.1.1 Einführung

Die meisten CCD-Kameras werden durch das sogenannte Lochkameramodell beschrieben [8, 31]. Dieses Modell ist aus dem physikalischen Aufbau der Kamera abgeleitet und besitzt den Vorteil, die Abbildung von 3D-Punkten der Szene auf die 2D-Bildebene mit Hilfe der projektiven Geometrie linear zu beschreiben. Durch diesen Ansatz können viele Probleme, wie beispielsweise die Kamerakalibrierung, mit Standardverfahren der linearen Algebra gelöst werden.
In den folgenden Abschnitten wird das Kameramodell näher erläutert und mathematisch formuliert.

### 3.1.2 Perspektivische Projektion

Eine 3D-perspektivische Projektion ist durch die Festlegung eines Projektionszentrums und einer 2D-Ebene vollständig definiert. Die Projektion eines 3D-Punktes ist durch den Schnittpunkt der Ebene mit der Linie, die durch das Projektionszentrum und den 3D-Punkt läuft, definiert.
Um eine mathematische Beschreibung zu ermöglichen wird ein Koordinatensystem eingeführt, dessen Ursprung im Projektionszentrum liegt. Die z-Achse liegt orthogonal zur

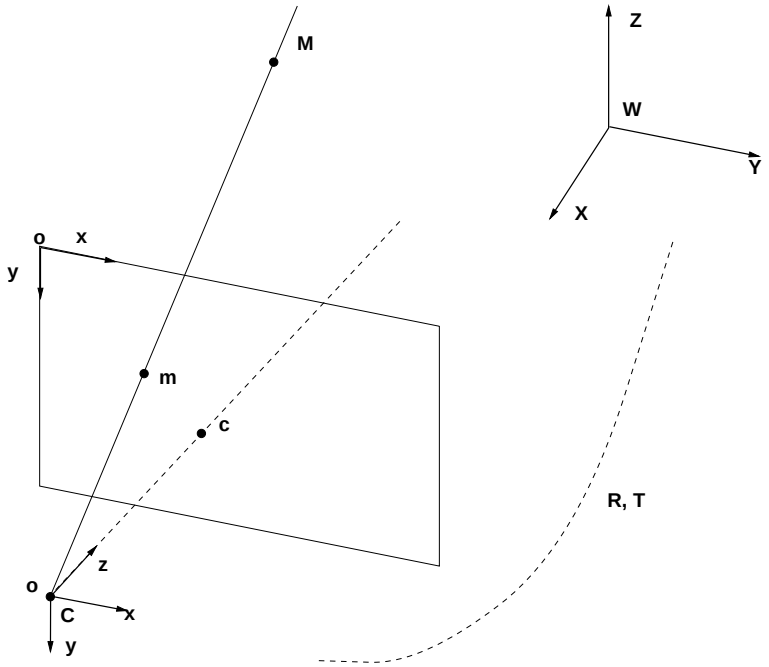



Abbildung 3.1: Kameramodell

Bildebene. Der Abstand zur Projektionsebene beträgt $z = 1$, und die Orientierung des Koordinatensystems um die $z$-Achse ist beliebig.
Das dazugehörige Koordinatensystem in der Ebene sei $(c, x_u, y_v)$, siehe Abbildung 3.1. Die Koordinaten des projiziertes Punktes $\boldsymbol{m}(x, y)$ von $\boldsymbol{M}(X, Y, Z)$ werden dann wie folgt beschrieben:

$$x = \frac{X}{Z} \tag{3.1}$$

$$y = \frac{Y}{Z} \tag{3.2}$$

Bei Einführung von homogenen Koordinaten können die obigen Gleichungen in Matrixform formuliert werden:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.3}$$

### 3.1.3 Intrinsische Parameter der Kamera

In Computer-Vision werden der Ursprung $\boldsymbol{c}$ und die z-Achse als *Bildhauptpunkt* und *optische Achse* der Kamera bezeichnet. Den Abstand vom Projektionszentrum zur Bildebene beschreibt die Brennweite $f$ der Kamera. Wenn $f = 1$ ist, spricht man von einer normierten Kamera. Das 3D-Koordinatensystem mit Ursprung im Projektionszentrum ist identisch mit dem Koordinatensystem der Kamera.

Um den projizierten Punkt im Bildkoordinatensystem beschreiben zu können, werden zusätzliche Parameter benötigt. $(c_x, c_y)$ sind die Koordinaten des Bildhauptpunktes $\boldsymbol{c}$ im Bildursprung $\boldsymbol{o}$, siehe Abbildung 3.1. Da die Pixel eines CCD Kamera-Chips nicht perfekt quadratisch sind, werden zwei zusätzliche Koeffizienten $s$ und $\alpha$ eingeführt. $s$ ist der Quotient der Pixelhöhe durch Pixelbreite. $\alpha$ beschreibt den Winkel, der die Abweichung zum senkrechten Winkel quantifiziert.
Nach Einführung dieser zwei Koeffizienten kann die Abbildung eines Punktes $\boldsymbol{m}_{\Re}(x_{\Re}, y_{\Re}, 1)$ im Koordinatensystem der Bildebene $\Re$ in einen Bildpunkt $\boldsymbol{m}(u, v, 1)$ im Bildkoordinatensystem wie folgt definiert werden:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fs & f\tan(\alpha) & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\Re} \\ y_{\Re} \\ 1 \end{pmatrix} \tag{3.4}$$

Dem entspricht:

$$\boldsymbol{m} = \boldsymbol{A}\boldsymbol{m}_{\Re} \tag{3.5}$$

Die Brennweite $f$ besitzt die Einheit Pixel pro Meter und der Bildhauptpunkt $\boldsymbol{c}(c_x, c_y)$ wird in Pixel beschrieben.
Die Matrix $\boldsymbol{A}$ ist eine Triangulärmatrix und beinhaltet die sogenannten *intrinsischen Kameraparameter*. In der Photogrammetrie-Literatur werden sie als die *innere Orientierung* der Kamera bezeichnet.

#### Mögliche Annäherungen

Für die meisten Kameras kann angenommen werden, dass die Pixel rechteckig sind, und der Winkel $\alpha$ folglich Null beträgt. Häufig wird auch die Approximation, dem "Aspect Ratio"$s$ gleich ein und der Bildhauptpunkt im Bildzentrum $\boldsymbol{c}$ zu setzen, verwendet.

### 3.1.4 Das Weltkoordinatensystem

Die Transformation eines Punktes $\boldsymbol{M}(X, Y, Z, 1)$ in $\boldsymbol{M}_c(X_c, Y_c, Z_c, 1)$ vom Welt- zum Kamera-Koordinatensystem wird wie folgt definiert:

$$\boldsymbol{M}_c = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}_3^{\top} & 1 \end{pmatrix} \boldsymbol{M} \tag{3.6}$$

wobei $\boldsymbol{R}$ und $\boldsymbol{t}$ die Rotation und die Translation vom Weltkoordinaten- zu Kamera-Koordinatensystem darstellen.
Die obige Gleichung kann auch folgendermaßen geschrieben werden:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \boldsymbol{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t} \tag{3.7}$$

Die Rotation $\boldsymbol{R}$ und die Translation $\boldsymbol{t}$ werden *externe Parameter* oder *äußere Orientierung* der Kamera genannt.

### 3.1.5 Die Projektionsmatrix

Ein Kameramodell wird vollständig durch die intrinsischen und externen Parametern beschrieben. Dabei gilt die Gleichung:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fs & 0 & c_{x_o} \\ 0 & f & c_{y_o} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.8}$$

Die Gleichung kann folgendermaßen formuliert werden:

$$\boldsymbol{m} \sim \boldsymbol{A}(\boldsymbol{R}, \boldsymbol{t})\boldsymbol{M} \sim \boldsymbol{PM} \tag{3.9}$$

wobei $\boldsymbol{P}$ eine $3 \times 4$-Matrix darstellt. $\boldsymbol{P}$ setzt sich aus den Komponenten der einzelnen Kameraparameter zusammen:

$$\boldsymbol{P} = \begin{pmatrix} f_x \boldsymbol{r}_1 + c_x \boldsymbol{r}_3 & f_x t_x + c_x t_z \\ f_y \boldsymbol{r}_2 + c_y \boldsymbol{r}_3 & f_y t_y + c_y t_z \\ \boldsymbol{r}_3 & t_z \end{pmatrix} \tag{3.10}$$

mit $\boldsymbol{r}_i (i = 1, 2, 3)$ als Zeilenvektor der Rotationsmatrix $\boldsymbol{R}$.

### 3.1.6 Definitionen

In diesem Abschnitt werden für Computer-Vision wesentliche Begriffe definiert.

**Definition 3.1 Äußere Orientierung:** *Die äußere Orientierung, kurz Kameraorientierung, erfasst den Aufnahmeort und die Aufnahmerichtung der Kamera.*

**Definition 3.2 Innere Orientierung:** *Die innere Orientierung bezeichnet die Parameter der Kamera selbst. Eine ideale Kamera wird vollständig durch die Brennweite $f$ und den Bildhauptpunkt $\boldsymbol{c}(cx, cy)$ beschrieben.*

**Definition 3.3 Kalibrierte Kamera:** *Eine Kamera wird als kalibriert bezeichnet, wenn die Matrix $\boldsymbol{A}$ der inneren Orientierung bekannt ist. In diesem Fall gilt für jeden Bildpunkt $\boldsymbol{m}$: $\boldsymbol{m} \sim \boldsymbol{A}^{-1}\boldsymbol{PM} \sim (\boldsymbol{R}, \boldsymbol{t})\boldsymbol{M}$*

**Definition 3.4 "Intersection" und "structure from motion":** *Die Begriffe "Intersection" und "structure from motion" bezeichnen die Berechnung der Szenestruktur aus einer bekannten Kamerabewegung.*

**Definition 3.5 Absolute Orientierung:** *Die absolute Orientierung beschreibt das Problem der Berechnung der inneren und äußeren Kameraorientierung aus Passpunkten.*

**Definition 3.6 Bundle adjustment:** *Bundle adjustment bezeichnet eine Optimierungsmethode, die Kameraorientierungen und Messungen in der Szene auf Basis einer Fehlerminimierung der Messungen in den Bildern verfeinert.*

## 3.2 Kamerakalibrierung

Die Kamerakalibrierung besitzt die Zielsetzung die innere Orientierung einer Kamera zu bestimmen. Dies erfolgt mit Hilfe eines sogenannten Kalibrierungsobjektes, das mit leicht zu detektierenden Marker versehen ist.
Die Koordinaten der Marker müssen sowohl im Bild als im Raum präzis bekannt d.h. eingemessen sein. Für den Markerpunkt $\boldsymbol{M}(X, Y, Z, 1)$ und die zugehörige Abbildung $\boldsymbol{m}(u, v, 1)$ gilt die folgende Projektionsmatrix $\boldsymbol{P}$:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.11}$$

gleichbedeutend mit der Gleichung:

$$m = \begin{pmatrix} \boldsymbol{p}_1 & p_{14} \\ \boldsymbol{p}_2 & p_{24} \\ \boldsymbol{p}_3 & p_{34} \end{pmatrix} \boldsymbol{M} \tag{3.12}$$

Bei Eliminierung des Skalierungsfaktors erhält man die sogenannte Kollinearitätsgleichung:

$$u = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{33}Y + p_{33}Z + p_{34}} \tag{3.13}$$

$$v = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{33}Y + p_{33}Z + p_{34}} \tag{3.14}$$

Für $n$ Punkte stehen $2 \times n$ Gleichungen zur Verfügung, die sich vereinfacht in Form eines linearen Gleichungssystems schreiben lassen:

$$\boldsymbol{AX} = 0 \tag{3.15}$$

mit $\boldsymbol{X} = (p_{11}, p_{12}, \ldots, p_{34})$.

Da $\boldsymbol{P}$ eine $3 \times 4$ Matrix abbildet und eine 2D/3D-Punktkorrespondenz zwei Gleichungen liefert, werden mindestens sechs Punkte benötigt, um das lineare System lösen zu können. Das Lösungsverfahren ist als DLT (Direct Linear Transformation) in der Literatur bekannt [84]. Grundsätzlich müssen bei dem DLT- Lösungsverfahren zusätzliche Einschränkungen eingeführt werden, um eine triviale Null-Lösung $\boldsymbol{X}$ zu vermeiden. Eine einfache Möglichkeit besteht darin z.B. $p_{34} = 1$ zusetzen und mit einem Least-Square (Pseudo-Inverse) das System zu berechnen [69]. Dieser Ansatz führt leider zu unstabilen und ungenauen Ergebnissen, wenn $p_{34}$ sich sehr stark von den anderen Elementen der $\boldsymbol{P}$ Matrix unterscheidet.
Das Faugeras-Toscani-Verfahren wertet eine Eigenschaft der Projektionsmatrix der Kamera aus, nämlich die Bedingung $\|\boldsymbol{p}_3\| = 1$. $\boldsymbol{p}_3$ entspricht dem Vektor $\boldsymbol{r}_3$ der Rotationsmatrix $\boldsymbol{R}$. Das Faugeras-Toscani-Verfahren liefert vergleichsweise stabilere Berechnungen als der vorherige beschriebene Ansatz.

### 3.2.1 Gewinnung der intrinsischen Parameter

Wenn $\boldsymbol{P}$ bekannt ist, können die intrinsischen Modellparameter bestimmt werden. Hierfür existieren zwei Möglichkeiten.
Wie im vorangehenden Abschnitt angemerkt ist, gilt: $\boldsymbol{p}_3 = \boldsymbol{r}_3$. Daraus folgt:

$$\begin{aligned} c_x &= \boldsymbol{p}_1\boldsymbol{p}_3 \\ c_y &= \boldsymbol{p}_2\boldsymbol{p}_3 \\ f &= \|\boldsymbol{m}_2 \times \boldsymbol{m}_3\| \\ fs &= -\|\boldsymbol{m}_1 \times \boldsymbol{m}_3\| \end{aligned} \tag{3.16}$$

$\boldsymbol{P}$ kann auch durch eine QR-Dekomposition aufgeteilt werden, wobei eine obere triangulare Matrix und eine orthonormale Matrix erhalten werden, denen die Matrizen $\boldsymbol{A}$ und $\boldsymbol{R}$ entsprechen.

### 3.2.2 Modellverfeinerung: Optische Verzerrung

Das vorgestellte Kameramodell stellt ein ideales, mathematisches Modell einer Lochkamera dar.
In der Praxis bilden jedoch die Linsensysteme keine lineare Projektion ab. Sie beinhalten nichtlineare Komponenten, die zu bedeutenden Verzerrungen in der Bildebene führen können. Insbesondere für Kameras mit Weitwinkel sind diese Verzerrungen offensichtlich und können bei dem Kalibrierungsprozess nicht außer Betracht gelassen werden. Eine Korrekturmöglichkeit besteht darin, die nichtlineare Deformation des Bildkoordinatensystems nachzubilden.
$\boldsymbol{m}(u, v)$ sei ein Bildpunkt einer realen Kamera und $\boldsymbol{c}(c_x, c_y)$ das Bildzentrum. Die korrigierten Koordinaten $(u', v')$ des Punktes $\boldsymbol{m}$ können dann wie folgt beschrieben werden:

$$\begin{aligned} u' &= u + k_1\bar{u}r^2 + k_2\bar{u}r^4 + k_3\bar{u}r^6 + P_1(2\bar{u}^2 + r^2) + 2P_2\bar{u}\bar{v} \\ v' &= v + k_1\bar{v}r^2 + k_2\bar{v}r^4 + k_3\bar{v}r^6 + P_2(2\bar{v}^2 + r^2) + 2P_1\bar{u}\bar{v} \end{aligned} \tag{3.17}$$

wobei $\bar{u} = u - u_o, \bar{v} = v - v_o, r = \bar{u}^2 + \bar{v}^2$ gilt.
Unterschiedliche Studien [85, 8] haben gezeigt, dass eine Modellierung der radialen Verzerrung für die meisten Anwendungen ausreichend genau ist und eine Modellierung bis zur ersten Ordnung $k_1$ bereits 90% der gesamten Verzerrung darstellt.

### 3.2.3 Nichtlineare Optimierung

Um die optischen Verzerrungsparameter zu berechnen, werden nicht-lineare Optimierungsverfahren angewendet. Dabei werden die Initialwerte der Kameraparameter durch eine lineare Lösung berechnet. Die Fehlerfunktion ist hierbei folgendermaßen definiert:

$$\sum_{i=1}^{n}(u_i - \frac{p_{i1}\boldsymbol{M}}{p_{i3}\boldsymbol{M}})^2 + (v_i - \frac{p_{i2}\boldsymbol{M}}{p_{i3}\boldsymbol{M}})^2 \tag{3.18}$$

## 3.3 Relative Orientierung I: Die fundamentale Matrix *F*

### 3.3.1 Einleitung

In diesem Abschnitt wird beschrieben, wie die relative Transformation, bzw. *relative Orientierung* und die Projektionsmatrizen zweier Kameras allein aus Bildern berechnen werden können.
Im ersten Schritt wird angenommen, dass keine Informationen über die Szene und die Kamera vorliegen. Es handelt sich hierbei um sogenannte Autokalibrierungsverfahren, für die ohne 3D-Kenntnisse die Kameraparameter bestimmt werden. Im zweiten Schritt wird erklärt, wie mit Hilfe der inneren Kameraparameter die Rotation $\boldsymbol{R}$ und die Translation $\boldsymbol{t}$ zwischen den Kamera berechnet werden kann.

### 3.3.2 Die epipolare Geometrie

Die fundamentale Matrix stellt die Matrix dar, die alle Informationen über zwei Bilder beinhaltet. Ihre robuste und genaue Schätzung ist deswegen für die Qualität der Autokalibrierungsmethode wesentlich.

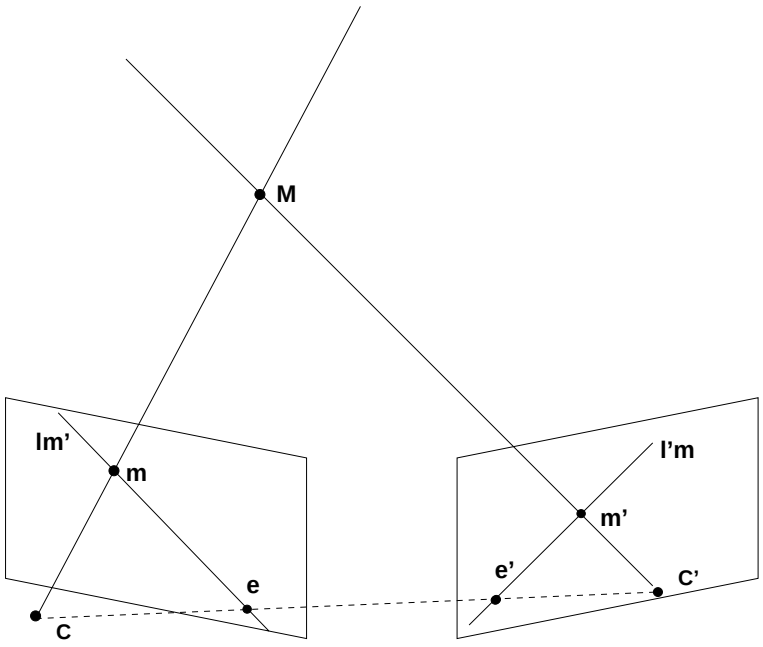



Abbildung 3.2: Epipolare Geometrie

$\boldsymbol{m}$ sei ein Punkt des ersten Bildes und $\boldsymbol{m'}$ der dazugehörigende Punkt im zweiten Bild (siehe Abbildung 3.2). Die fundamentale Matrix $\boldsymbol{F}$ ist dann wie folgt definiert:

$$\boldsymbol{m'Fm} = 0 \tag{3.19}$$

### 3.3.3 Der Acht-Punkte-Algorithmus

Der Acht-Punkte-Algorithmus stellt einen grundlegenden Algorithmus im Bereich Computer-Vision dar. $n$ ($n >= 8$) Punktpaare, $(\boldsymbol{m}_i, \boldsymbol{m}'_i)$ für $(i = 1, \ldots, n)$ werden mit Hilfe der

sogenannten Longuet-Higgins-Gleichung [57] beschrieben und die Matrix $\boldsymbol{F}$, die die quadratische Fehlerfunktion $Q(\boldsymbol{F})$ minimiert, berechnet.
$Q$ ist wie folgt definiert:

$$Q(\boldsymbol{F}) = \sum_{i=1}^{n} \|\boldsymbol{m}_i' \boldsymbol{F} \boldsymbol{m}_\mathtt{i}\|^2 \tag{3.20}$$

$\boldsymbol{f}$ sei ein 9-dimensionaler Vektor, der die Matrixelemente von $\boldsymbol{F}$ beinhaltet:

$$\boldsymbol{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}) \tag{3.21}$$

Für ein gegebenes Punktpaar $\boldsymbol{m}_i(u_i, v_i, 1)$ und $\boldsymbol{m}_i'(u_i', v_i', 1)$ gilt dann:

$$\boldsymbol{m}_i' \boldsymbol{F} \boldsymbol{m}_\mathtt{i} = \boldsymbol{a}_\mathtt{i} \boldsymbol{f} \tag{3.22}$$

mit

$$\boldsymbol{a}_i = (u_i u_i', v_i u_i', u_i, u_i v_i', v_i v_i', v_i, u_i, v_i, 1) \tag{3.23}$$

Wenn man eine $n \times 9$ Matrix $\boldsymbol{A}$ mit dem Vektor $\boldsymbol{a}_i$ in der Zeile $i$ eingeführt, ergibt sich aus den Gleichungen 3.20 und 3.22:

$$Q(\boldsymbol{F}) = \sum_{i=1}^{n} \|\boldsymbol{a}_i . \boldsymbol{f}\|^2 = \|\boldsymbol{A}\boldsymbol{f}\|^2 \tag{3.24}$$

Um die triviale Lösung $\boldsymbol{f} = \boldsymbol{0}$ zu vermeiden, wird $\|\boldsymbol{f}\|^2 = 1$ gesetzt.
Eine SVD-Dekomposition wird auf die Matrix $\boldsymbol{A}$ angewendet. Daraus folgt:

$$Q(\boldsymbol{F}) = \|\boldsymbol{A}\boldsymbol{f}\|^2 = \|\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}\boldsymbol{f}\|^2 = \|\boldsymbol{S}\boldsymbol{V}\boldsymbol{f}\|^2 >= \sigma_9^2 \|\boldsymbol{V}\boldsymbol{f}\|^2 >= \sigma_9^2 \|\boldsymbol{f}\|^2 \tag{3.25}$$

mit der diagonalen Matrix $\boldsymbol{S} = Diag(\sigma_1, \ldots, \sigma_9)$ der Singularwerte $\sigma_i$ mit $i = 1, \ldots, n$.
Zusätzlich wird an das Kriterium $Q$ die Bedingung $\|\boldsymbol{f}\| = 1$ gestellt:

$$Q(\boldsymbol{F}) >= \sigma_9^2 \tag{3.26}$$

Diese Gleichung(3.26) besagt, dass das Minimum der Funktion $Q$ für den Vektor $\boldsymbol{V}\boldsymbol{e}_9$ mit $\boldsymbol{e}_9 = (000000001)$ erreicht ist.

### Numerische Stabilität

Das oben vorgestellte Verfahren verhält sich instabil, wenn die Daten nicht vorher im Intervall $[-\sqrt{2}; \sqrt{2}]$ normiert worden sind. Weitere mögliche Transformationen der Bilddaten sind in [47, 65]zu finden. Mit Hilfe einer passenden Transformation kann eine ähnliche Präzision mit linearen Verfahren wie mit nichtlinearen Verfahren erreicht werden.

### Rank Constraint

Da für die Epipole $\boldsymbol{e}$ und $\boldsymbol{e}'$, $\boldsymbol{F}\boldsymbol{e} = 0$ und $\boldsymbol{F}\boldsymbol{e}' = \mathtt{0}$ gilt, beträgt der Rank von $\boldsymbol{F}$ zwei oder weniger. Auf Grund von Ungenauigkeiten folgt die mit dem Acht-Punkte-Algorithmus gefundene Matrix die Rankeigenschaft der fundamentale Matrix nicht. Diese Eigenschaft kann durch die Bestimmung der nahliegensten Matrix mit Rank 2 erzwungen werden [37]. Die Matrix kann durch $\boldsymbol{F} = \boldsymbol{U}D(w_1, w_2, w_3)\boldsymbol{V}$ beschrieben werden. Die nahliegensten Matrix mit Rank zwei kleinste Singularwert wird hierfür auf Null gesetzt: $\boldsymbol{F} = \boldsymbol{U}D(w_1, w_2, 0)\boldsymbol{V}$.

### 3.3.4 Nichtlineare Optimierung

Die Minimierung der Distanz zwischen den Punkten $\boldsymbol{m}'$ und ihrer epipolaren Linie $l'_m$ stellt eine wesentliches Kriterium für die Optimierung von $\boldsymbol{F}$ dar. Die Distanz $d(\boldsymbol{m}', \boldsymbol{l}'_m)$ kann folgendermaßen berechnet werden:

$$d(\boldsymbol{m}', \boldsymbol{l}'_m) = \frac{\boldsymbol{m}'^{\top} \boldsymbol{l}'_m}{\sqrt{l'^2_{1m} + l'^2_{2m}}} = \frac{1}{c_m} \boldsymbol{m}' \boldsymbol{F} \boldsymbol{m} \tag{3.27}$$

mit $c_m = \sqrt{l'^2_{1m} + l'^2_{2m}}$ und $\boldsymbol{l}'_m = (l'_{1m}, l'_{2m}, l'_{3m})$
Das Kriterium kann dann für alle Punkte angesetzt werden, wobei folgende Gleichung gilt:

$$\sum_{i=1}^{n} (d(\boldsymbol{m}_{i2}, \boldsymbol{l}_{m'_{i1}})^2 + d(\boldsymbol{l}_{m'_{i2}, \boldsymbol{m}_{i1}})^2) \tag{3.28}$$

dem entspricht:

$$\sum_{i=1}^{n} (\frac{1}{l^2_{1m} + l^2_{1m}} + \frac{1}{l^2_{1m'_i} + l^2_{1m'_{i1}}})(\boldsymbol{m}_2^{\top} \boldsymbol{F} \boldsymbol{m}_1)^2 \tag{3.29}$$

In der Literatur werden weitere Optimierungskriterien vorgestellt. Eine umfangreiche Analyse und ein Vergleich der verschiedenen Methoden können beispielsweise in [91] gefunden werden.

### 3.3.5 Parametrisierung

Die Matrix $\boldsymbol{F}$ ist eine singuläre Matrix ($det(\boldsymbol{F}) = 0$) und besitzt Rank zwei. Die folgende Parametrisierung [71, 32] berücksichtigt die Bedingung, dass die Anzahl der unabhängigen Koeffizienten von $\boldsymbol{F}$ sieben (7 Freiheitsgrade) beträgt:

$$\begin{pmatrix} b & a & -ay - bx \\ -d & -c & cy + dx \\ dy' & cy' - ax' & -cyy' - dy'x + ayx' + bxx' \end{pmatrix} \tag{3.30}$$

### 3.3.6 Die Epipole $\boldsymbol{e}$ und $\boldsymbol{e}'$

### 3.3.7 Faktorisierungsmethode

Wenn $\boldsymbol{F}$ die fundamentale Matrix zweier Kameras abbildet, kann $\boldsymbol{F}$ durch das Produkt von $[\boldsymbol{e}']_\wedge$ und einer Matrix $\boldsymbol{M}$, d.h. $\boldsymbol{F} = [\boldsymbol{e}']_\wedge \boldsymbol{M}$, beschrieben werden. Die Projektionsmatrizen $\boldsymbol{P}$ und $\boldsymbol{P}'$ der Kameras sind hierbei folgendermaßen definiert:

$$\boldsymbol{P} = [\boldsymbol{I}, \boldsymbol{0}]$$
$$\boldsymbol{P}' = [\boldsymbol{M}, \boldsymbol{e}']$$

In dem Fall, dass $\boldsymbol{P}$ und $\boldsymbol{P}'$ gegeben sind, kann aus ihnen einfach die Matrix $\boldsymbol{F} = [\boldsymbol{e}']_\wedge \boldsymbol{M}$ berechnet werden. Außerdem gilt, wenn $\boldsymbol{M}$ eine Lösung ist, dass $\boldsymbol{M} + \boldsymbol{e}' \boldsymbol{v}^{\top}$ auch eine Lösung von $\boldsymbol{F}$, für alle beliebigen Vektoren $\boldsymbol{v}$ (denn $[\boldsymbol{e}']_\wedge \boldsymbol{e}' \boldsymbol{v}^{\top} = \boldsymbol{0}$), liefert.

Eine Faktorisierungstechnik von $\boldsymbol{F}$ kann in [60] gefunden werden. Dabei wird folgende Eigenschaft angewendet:

$$\|\boldsymbol{v}\|^2\boldsymbol{I}_3 = \boldsymbol{v}\boldsymbol{v}^\top - [\boldsymbol{v}]_\wedge^2 \tag{3.31}$$

Für die fundamentale Matrix $\boldsymbol{F}$ gilt dann:

$$\boldsymbol{F} = \frac{1}{\|\boldsymbol{e}\|^2}(\boldsymbol{e}'\boldsymbol{e}'^\top - [\boldsymbol{e}']_\wedge^2)\boldsymbol{F} = \frac{1}{\|\boldsymbol{e}\|^2}\boldsymbol{e}'\boldsymbol{e}'^\top\boldsymbol{F} + [\boldsymbol{e}']_\wedge(-\frac{[\boldsymbol{e}']_\wedge}{\|\boldsymbol{e}\|^2}\boldsymbol{F}) \tag{3.32}$$

Der erste Term ist gleich null, da $\boldsymbol{F}\boldsymbol{e}' = 0$. Die Matrix $\boldsymbol{M}$ ist gegeben durch $\boldsymbol{M} = -\frac{[\boldsymbol{e}']_\wedge}{\|\boldsymbol{e}\|^2}\boldsymbol{F}$. Numerisch werden bessere Ergebnisse für die 3D-Rekonstruktion erhalten, wenn der Epipol $\boldsymbol{e}$ normalisiert wird [46].

## 3.4 Relative Orientierung II: Die essentielle Matrix $\boldsymbol{E}$

### 3.4.1 Einleitung

Die essentielle Matrix $\boldsymbol{E}$ verbindet die Punkte aus zwei kalibrierten Ansichten durch:

$$\boldsymbol{m}'^\top\boldsymbol{E}\boldsymbol{m} = 0 \tag{3.33}$$

mit $\boldsymbol{E} = [\boldsymbol{t}]_\wedge\boldsymbol{R}$.
Zwischen der Matrix $\boldsymbol{F}$ und der Matrix $\boldsymbol{E}$ existiert folgende Beziehung:

$$\boldsymbol{E} = \boldsymbol{A}_2^\top\boldsymbol{F}\boldsymbol{A}_1 \tag{3.34}$$

wobei $\boldsymbol{A}_1$ und $\boldsymbol{A}_2$ die Matrizen der intrinsischen Parameter beider Kamera darstellen.

#### Eigenschaften

Die Eigenschaften der Matrix $\boldsymbol{E}$ werden durch ihren Rank und ihre Eigenwerte beschrieben. In [48] beweisen die Autoren unter welchen Bedingungen eine $3\times3$ Matrix in eine Rotation $\boldsymbol{R}$ und Translation $\boldsymbol{t}$ zerlegt werden kann, d.h. wann eine Matrix eine essentielle Matrix darstellt.

**Theorem 3.1** *Eine $3 \times 3$ Matrix $\boldsymbol{E}$ ist eine essentielle Matrix, wenn und nur wenn die singulären Werte $\sigma_1 >= \sigma_2 >= \sigma_3$ die folgende Bedingung aufweisen: $\sigma_1 = \sigma_2 > 0$ und $\sigma_3 = 0$*

Der Theorembeweis kann in beispielsweise in folgenden Veröffentlichungen [48, 31, 50] nachgelesen werden.

### 3.4.2 Bestimmung von $\boldsymbol{E}$

Die Matrix $\boldsymbol{E}$ hängt von fünf Parametern ab und ein Punktpaar $(\boldsymbol{m}, \boldsymbol{m}')$ liefert eine Gleichung für die Berechnung von $\boldsymbol{E}$, siehe Gleichung (3.33). Aus diesem Grund sind fünf Bildpunktpaare ausreichend, um $\boldsymbol{E}$ zu bestimmen. Eine Lösung dafür wird in [31] erläutert. Die Ergebnisse sind jedoch instabil und hängen direkt von der Genauigkeit der Eingabedaten ab. Deswegen wird meistens auf den "Standard"-Acht-Punkte-Algorithmus zurückgegriffen.

### 3.4.3 Bestimmung von $\boldsymbol{R}$ und $\boldsymbol{t}$

Nach der Berechnung der Matrix $\boldsymbol{E}$ muss sicher gestellt werden, dass $\boldsymbol{E}$ die Eigenschaften einer essentiellen Matrix besitzt.
Hierfür wird eine SVD-Zerlegung von $\boldsymbol{E}$ vorgenommen, und die theoretischen Werte der Singularwerte werden erzwungen:

$$\boldsymbol{E} = \boldsymbol{U}\boldsymbol{D}(1,1,0)\boldsymbol{V} \approx \boldsymbol{U}\boldsymbol{D}(\mathtt{w}_1,\mathtt{w}_2,\mathtt{w}_3)\boldsymbol{V} \tag{3.35}$$

In diesem Fall gelten die Singularwerte $(w_1, w_2, w_3)$ als Indikator der Stabilität der Berechnungen des Gleichungssystems. Die Ratios $\frac{\sigma_1}{\sigma 2}$ und $\frac{\sigma_1}{\sigma 3}$ sollen einen zu den theoretischen ähnliche Wert aufweisen, d.h. in etwa den Wert Eins und Null betragen.
Es gibt verschiedene Methoden, um die relative Orientierung $\boldsymbol{R}$ und aus $\boldsymbol{E}$ zu berechnen [86, 5]. An dieser Stelle wird auf ein robustes Verfahren, dass auf SVD basiert [48], verwiesen. Die Ergebnisse dieses Verfahrens werden im folgenden vorgestellt.
Wenn als Zerlegung der Matrix $\boldsymbol{E}$ sei $\boldsymbol{E} = \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^\top$ definiert wird, lassen sich die zwei Rotationsmatrizen $\boldsymbol{R}_a$ und $\boldsymbol{R}_b$ aus $\boldsymbol{E}$ wie folgt abgeleiten:

$$\boldsymbol{R}_a = \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^\top \tag{3.36}$$

und

$$\boldsymbol{R}_b = \boldsymbol{U}\boldsymbol{W}^\top\boldsymbol{V}^\top \tag{3.37}$$

mit

$$\boldsymbol{W} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.38}$$

Die Berechnung von $\boldsymbol{t}$ ist dann relativ einfach, da $\boldsymbol{E}\boldsymbol{t} = \mathtt{0}$ gilt. Der Vektor $\boldsymbol{t}$ stellt den Eigenvektor mit dem kleinsten Eigenwert (Null) von $\boldsymbol{E}$ dar.
Letztendlich existieren zwei Lösungen $(\boldsymbol{R}_a, \boldsymbol{t})$ und $(\boldsymbol{R}_b, \boldsymbol{t})$ für $\boldsymbol{E}$, die in der Literatur als "twisted pair" bezeichnet werden [48].
Für die Epipole von $\boldsymbol{E}$ lassen sich folgende Gleichungen aufstellen: $\boldsymbol{e'} = \boldsymbol{t}$ und: $e = -\boldsymbol{R}^{-1}\boldsymbol{t}$

### 3.4.4 Planare Szene

### 3.4.5 Homographieberechnung

Eine Homographie stellt eine 2D-Kollineation dar, d.h. eine lineare Transformation, deren Transformationsmatrix nicht singulär ist.
Für jeden Bildpunkt $\boldsymbol{m}$ im ersten Bild und $\boldsymbol{m'}$ im zweiten Bild ein und derselben Ebene existiert hierbei eine $3 \times 3$ Matrix, so dass geschrieben werden kann: $\boldsymbol{m'} = \boldsymbol{H}\boldsymbol{m}$ und $det(\boldsymbol{H})! = 0$.
$\boldsymbol{H}$ ist bis zu einem Skalierungsfaktor definiert, d.h. $\boldsymbol{H}$ hängt von acht unabhängigen Koeffizienten ab. Mit Hilfe von vier oder mehr Punkten kann $\boldsymbol{H}$ linear bestimmt werden.

### 3.4.6 Verhältnisse zwischen 2D-Homographie und fundamentaler Matrix

Für die Punkte $\boldsymbol{m}$ im ersten und $\boldsymbol{m'}$ im zweiten Bild, die gleichzeitig ein und derselben Ebene angehören, gelten die folgenden Gleichungen:

$$\boldsymbol{m'}^\top\boldsymbol{F}\boldsymbol{m} = 0 \tag{3.39}$$

und:

$$\boldsymbol{m}' = \boldsymbol{H}\boldsymbol{m} \tag{3.40}$$

Daraus ergibt sich:

$$\boldsymbol{m}\boldsymbol{H}^\top\boldsymbol{F}\boldsymbol{m} = 0 \tag{3.41}$$

Die Gleichung (3.41) ist für jeden beliebigen Punkt $\boldsymbol{m}$ gültig, d.h. , dass die Matrix $\boldsymbol{H}^\top\boldsymbol{F}$ antisymmetrisch ist. Sie verifiziert deswegen die folgende Eigenschaft [71]:

$$\boldsymbol{H}^\top\boldsymbol{F} + \boldsymbol{F}^\top\boldsymbol{H} = 0 \tag{3.42}$$

Darüber hinaus gilt:

$$\boldsymbol{F}\boldsymbol{m} = \boldsymbol{l}'_m = \boldsymbol{e}'\wedge\boldsymbol{m}' \tag{3.43}$$

gleichbedeutend zu:

$$\boldsymbol{m}'\boldsymbol{F}\boldsymbol{m} = \boldsymbol{m}'[\boldsymbol{e}']_\wedge\boldsymbol{m}' = \boldsymbol{m}'[\boldsymbol{e}']_\wedge\boldsymbol{H}\boldsymbol{m} = 0 \tag{3.44}$$

Zusammenfassend kann die Beziehung zwischen $\boldsymbol{F}$ und $\boldsymbol{H}$ durch die folgende Gleichung beschrieben werden:

$$\boldsymbol{F} = [\boldsymbol{e}']_\wedge\boldsymbol{H} \tag{3.45}$$

Aus dieser Gleichung kann abgeleitet werden, dass $\boldsymbol{H}\boldsymbol{e} = \boldsymbol{e}'$ für jede beliebige Ebene gilt, und dass für eine Ebene, die durch das optische Kamerazentrum $\boldsymbol{c}$ verläuft, $\boldsymbol{H}\boldsymbol{e} = 0$ folgt. Als weitere charakteristische Eigenschaft sei genannt, dass durch die Homographie $\boldsymbol{H}^\top$ die beiden epipolaren Linien zweier Bilder wie folgt in Bezug gesetzt werden:

$$\boldsymbol{H}^\top\boldsymbol{l}' = \boldsymbol{H}^\top\boldsymbol{F}\boldsymbol{m} \sim \boldsymbol{F}^\top\boldsymbol{H}\boldsymbol{m} = \boldsymbol{F}^\top\boldsymbol{m}' = \boldsymbol{l} \tag{3.46}$$

### 3.4.7 Faktorisierungsmethode

Wenn die Kameras intern kalibriert sind, kann man ihre relative Position der Kameras zueinander und die Kalibrierungsebene bestimmen [Wunderlich 1982] [31]. Das Verfahren führt zu zwei Lösungen, wobei die richtige Lösung mit einem sogenannten *visibility test* kann bestimmen werden.

## 3.5 Rekonstruktion

Die Rekonstruktion von 3D-Punkten der Szene ist zur Erstellung eines 3D-Modells für die Verdeckungsbehandlung virtueller und realer Objekte notwendig.
In diesem Abschnitt wird einleitend als erster Schritt der Rekonstruktion die Tiefenberechnung aus zwei Bildern erläutert.

### 3.5.1 Tiefenberechnung

$\boldsymbol{m}$ sei ein Punkt des ersten Bildes und $\boldsymbol{m'}$ der dazugehörigende Punkt im zweiten Bild. Die Tiefe der Punkte wird jeweils mit $z$ und $z'$ bezeichnet, so dass folgende Beziehung gilt:

$$z\boldsymbol{m} = z'\boldsymbol{R}\boldsymbol{m'} + \boldsymbol{t} \tag{3.47}$$

Eine direkte Berechnung von $z$ und $z'$ ist sehr rauschempfindlich. Aus diesem Grund wird ein Verfahren zur Fehlerquadratminimierung, das folgenderweise definiert ist, vorgezogen:

$$\|(z\boldsymbol{m} - z'\boldsymbol{R}\boldsymbol{m'}) - \boldsymbol{t}\|^2 \rightarrow min. \tag{3.48}$$

Mit Hilfe der Ableitung der obigen Gleichung (3.48) in $z$ und in $z'$, werden die Minima bestimmt. Sie sind gegeben durch:

$$z = \frac{\boldsymbol{t}\cdot\boldsymbol{m} - (\boldsymbol{m}\cdot\boldsymbol{R}\boldsymbol{m'})(\boldsymbol{t}\cdot\boldsymbol{R}\boldsymbol{m'})}{1 - \boldsymbol{m}\cdot\boldsymbol{R}\boldsymbol{m'}} \tag{3.49}$$

und

$$z' = \frac{(\boldsymbol{t}\cdot\boldsymbol{m})(\boldsymbol{m}\cdot\boldsymbol{R}\boldsymbol{m'}) - \boldsymbol{t}\cdot\boldsymbol{R}\boldsymbol{m'}}{1 - \boldsymbol{m}\cdot\boldsymbol{R}\boldsymbol{m'}} \tag{3.50}$$

### 3.5.2 Lineare Lösung

Es seien $n$ Bilder und deren Projektionsmatrizen $(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n)$ mit $(i = 1, \ldots, n)$ gegeben. Der zu rekonstruierende 3D-Punkt $\boldsymbol{M}(X, Y, Z, 1)$ wird im Bild $i$ in $\boldsymbol{m}_i(u_i, v_i, 1)$ projiziert. Es gilt dann:

$$\boldsymbol{m}_i = \boldsymbol{P}_i\boldsymbol{M} \tag{3.51}$$

Aus der Gleichung 3.51 resultiert ein Gleichungssystem:

$$\boldsymbol{A}\boldsymbol{M} = 0 \tag{3.52}$$

mit

$$\boldsymbol{A} = [p_{10} - u_1 p_{10}, p_{11} - v_1 p_{12}, \ldots, p'_{n0} - u_n p'_{n2}, p'_{n1} - v_n p'_{n2}] \tag{3.53}$$

$\boldsymbol{A}$ stellt eine $2n \times 4$ Matrix dar. Das Gleichungssystem kann mit einem SVD-Verfahren gelöst werden.

### 3.5.3 Minimierung der Reprojektionsfehler

Die Minimierung der Reprojektionsfehler über allen Punkte wird wie folgt definiert:

$$\sum_{i=1}^{n}(u_i - \frac{p_{i0}\mathtt{M}}{p_{i2}\mathtt{M}})^2 + (v_i - \frac{p_{i1}\mathtt{M}}{p_{i2}\mathtt{M}})^2 \tag{3.54}$$

Diese Gleichung ist äquivalent zu

$$D_i^{-1}(u_i p_{i2} - p_{i0})\boldsymbol{M} = \mathtt{0} \tag{3.55}$$

und

$$D_i^{-1}(v_i p_{i2} - p_{i1})\boldsymbol{M} = \mathtt{0} \tag{3.56}$$

wobei für die Wichtung $D_i = p_{i2}^{\top}$ gilt; $D_i$ wird in der ersten Iteration mit 1 gewichtet.

## 3.6 Zusammenfassung

In diesem Kapitel wurden die Grundlagen der Computer-Vision vorgestellt. Nach der Erläuterung des Lochkameramodells, wurden die Prinzipen der Kamerakalibrierung beschrieben. Anschließend erfolgte eine Betrachtung der aus zwei Bildern resultierenden Kamerageometrie. Insbesondere wurde gezeigt, in welcher Weise die Projektionsmatrizen der Kamera im projektiven Raum mit Hilfe der fundamentalen Matrix zurückgewonnen werden können. Die Bestimmung der relativen Bewegungen zwischen zwei Bildern erfolgte auf Basis der essentielle Matrix bzw. der Homographie für planare Szenen. Abschließend wurden Methoden der 3D-Rekonstruktion an Hand eines linearen und eines iterativen Verfahrens erläutert.

# Kapitel 4

# Augmented Images und Videos

In dem Kapitel "Augmented Images und Videos" wird zunächst die Kernproblematik von Augmented-Reality, d.h. die Erweiterung mit virtuellen Objekten einzelner Bilder, behandelt. Anschließend werden Verfahren zur Bearbeitung kompletter Videoabfolgen entwickelt und evaluiert.
Die erste Fragestellung des Erweiterungsprozesses betrifft die Festlegung von Position, Orientierung und Skalierung des virtuellen Objektes in seiner realen Umgebung. Als schwierig erweist sich hierbei die Tatsache, dass meist als einzige Informationsquelle nur Bilder vorliegen und wenige bis gar keine 3D-Daten der Szene zur Verfügung stehen.
Die zweite Fragestellung bezieht sich auf die Bestimmung der jeweiligen Kameraparameter, die eine lagerichtige Projektion der virtuellen Objekte in die Bilder ermöglichen. Die Lösungen müssen leicht einsetzbar sein, wobei möglichst wenige Bildangaben oder externe Informationen, wie z.B. 3D Koordinaten von Szenenpunkten, vorausgesetzt werden sollten.
Nach der Definition des neuen Begriffes *Augmented Image* wird ein komplettes Kalibrierungsverfahren einer Kamera vorgestellt. Anschließend werden die Erweiterungsmöglichkeiten von zunächst einem auf mehrere Bilder betrachtet und neue flexible Verfahren entwickelt. Automatisierungsmechanismen, wie beispielsweise die automatische Punktverfolgung, werden für die Bearbeitung von Bildfolgen eingeführt. Zum Schluss erfolgt eine Diskussion der vorgeschlagenen Lösungen und eine Analyse ihrer Einsetzmöglichkeiten in der Praxis.

## 4.1 Augmented Images

### 4.1.1 Definition

Die Off-line-Erweiterung eines Bildes mit virtuellen Objekten stellt eine abgeleitete Form von AR dar. Der vollständige Bezug zur Realität, wie beispielsweise die natürliche Wahrnehmung der Umgebung, ist hierbei nicht mehr gegeben.
Aus diesem Grund wird der neue Begriff *Augmented Image* (AI) eingeführt und als die Off-line-Erweiterung eines Bildes der realen Welt mit 3D-Objekten definiert.
Eine 3D Registrierung des virtuellen Objektes mit der Szene wird vorausgesetzt und es wird angenommen, dass die Kameraposition und -orientierung und der Abbildungsprozess auf die Bildebene mathematisch modelliert und berechnet worden sind. Durch diesen letzten Punkte unterscheidet sich ein AI von einer klassischen Fotomontage und ermöglicht

beispielsweise eine geometrisch korrekte Behandlung der Verdeckung virtueller Objekte oder eine physikalisch basierte Lichtsimulation des erweiterten Bildes.

### 4.1.2 Bildung eines Augmented Images

Die Bildung eines AI wird in die drei folgenden Schritten untergeteilt:

1. Registrierung des virtuellen Objektes in die 3D Szene

   Der erste Schritt beinhaltet die geometrische Registrierung des virtuellen Objektes in der realen Szene. Die Position, Orientierung und Skalierung bezüglich der realen Umgebung werden hierbei festgelegt.

(a) (b) (c)

Abbildung 4.1: (a) 2D-Sicht der realen, (b) virtuelle Welt und (c) Augmented-Image

   Diese Transformation wird als Transformation $\boldsymbol{T}$ des Koordinatensystems des virtuellen Objektes zu dem Koordinatensystem der realen Umgebung definiert. Es handelt sich dabei um eine Similaritätstransformation, die von sieben Parameter (3 Rotationswinkel, 3 Positionskoordinaten, 1 Skalierung) abhängt.

   Für diese Transformation muss vorrausgesetzt sein, dass die reale Umgebung bereits modelliert, d.h. ein Referenzkoordinatensystem festgelegt und die Szene, zumindest teilweise, nachgebildet wurde.

2. Registrierung der Kamera

   Der zweite Schritt besteht in der Registrierung der Kamera in der realen Umgebung. Durch diesen Schritt wird der Abbildungsprozess der realen Szene auf das zu bearbeitende Bild charakterisiert.

   Wie im Kapitel 3 erläutert wurde, wird die entsprechende Abbildung durch eine Projektionsmatrix $\boldsymbol{P}$ bestimmt. Diese Matrix setzt sich aus Position, Orientierung sowie intrinsischen Parametern der Kamera zusammen. Die genannten Parameter werden dann bei dem Rendering-System so eingestellt, dass die dem Bild der realen Szene entsprechende Ansicht nachgebildet werden kann.

   Meistens stellt die Berechnung der Kameraparameter das größte Problem der Erzeugung eines AIs dar. Die Projektionsmatrix einer Kamera kann aus der Zuordnung von 2D-Bildmerkmalen mit 3D-Punkten der Szene zurückgewonnen werden. Dennoch ist das Problem "ill-conditionned", siehe Kapitel **??**, [31] und die Berechnungen numerisch instabil. In der Praxis können bessere Ergebnisse erreicht werden, indem die Aufnahmen mit einer kalibrierten Kamera realisiert werden. Dadurch müssen

weniger Parameter bestimmt werden und somit kann eine höhere Stabilität der Berechnungen erzielt werden.

Ein vollständiges System zur genauen Kamerakalibrierung, das sowohl die Bildverarbeitungsprozesse als auch die Kalibrierungsalgorithmen berücksichtigt, wird im Abschnitt 4.2 vorgestellt. Ein praktisches System zu Erzeugung von Augmented Images wird im Abschnitt 4.3.4 erläutert.

3. Bildsynthese

   Der letzte Schritt, die Bildsynthese, beschäftigt sich mit dem Rendering der virtuellen Objekte und des ursprünglichen Bildes der realen Szene. An dieser Stelle stehen vor allem Rendering-Aspekte im Vordergrund, die aus dem gegenseitigen Einfluss realer und virtueller Objekte resultieren.

   Hierbei handelt es sich einerseits um die Verdeckungsbehandlung zwischen realen und virtuellen Objekten und andererseits um die Berechnung der neuen Lichtverhältnisse, wie beispielsweise Schatten oder Lichtreflexionen. Eine konsistente Bildsynthese ist wichtig, da sie die Wahrnehmung des erweiterten Bildes unterstützt.

   In beiden Fällen wird ein 3D-Modell der realen Szene benötigt. In der Praxis stehen leider meist nur sehr wenige 3D-Informationen der betreffenden Umgebung zur Verfügung, und die Durchführung von Vermessungen ist zeitintensiv und kostenspielig. Auch in diesem Bereich müssen neue Ansätze, die nur auf 2D-Daten aus Bildern beruhen, entwickeln werden.

### 4.1.3 Vorgehensweise

In den folgenden Abschnitten werden systematische Methoden zur Erzeugung eines AIs vorgestellt. Dies beinhaltet

1. die Bestimmung der Kamera-Position und -orientierung
2. die Erzeugung eines Szenemodells für die Verdeckungsbehandlung aus den Bildern
3. und die Platzierung des virtuellen Objektes in seiner realen Umgebung.

Da die Bestimmung der inneren Kameraparameter den ersten Arbeitsschritt darstellt, wird einleitend eine Methode zur präzisen Kamerakalibrierung vorgestellt.

## 4.2 Kamerakalibrierung

### 4.2.1 Einleitung

In einem Kameramodell stecken verschiedene physikalische und geometrische Parameter, siehe Kapitel 3. Die Bestimmung dieser Kameraparameter wird Kamerakalibrierung genannt. Die Kalibrierung erfolgt durch die Aufnahme eines Kalibrierungsmusters, welches Referenzpunkte in der 3D-Welt bereitstellt. Diese Referenzpunkte bilden mit ihren Bildpunkten zusammen die Grundlage zur Berechnung der Kameraparameter.
Entscheidend für die Güte der geschätzten Parameter ist die sub-pixel genaue Bestimmung der Kalibrierungspunkte in der Aufnahme.

Die Genauigkeit der Modellabbildung ist wesentlich bei AR-Anwendungen, da sie direkt die Qualität der Überlagerung beeinflusst. Zur Zeit konzentrieren sich viele in der Literatur zu findende Arbeiten nur auf einzelne Aspekte der Kamerakalibrierung. Selten, wie z.B. in [83, 49], wird dabei auf die gesamte Vorgehensweise, d.h. die Merkmalextraktion und die Berechnung der $\boldsymbol{P}$-Matrix, die Bildverzerrungskorrektur und Fehlerschätzung einbezieht, eingegangen. Um bei der Bildüberlagerung für AR präzise Ergebnisse zu erreichen, muss aber das Zusammenspiel jeder Komponente berücksichtigt werden. Auf Grund dessen wurde im Rahmen dieser Arbeit ein komplettes System "MAXCAL", das sowohl die Bildverarbeitung als auch die Kalibrierungsalgorithmen beinhaltet, erarbeitet.
Im folgenden werden alle Schritte der Kamerakalibrierung, d.h. die Lokalisierung der Kalibrierungsmarker, deren präzise Extraktion, das Kalibrierungsalgorithmus und die Evaluierung des gesamten Systems, im Detail vorgestellt.

### 4.2.2 Bildverarbeitungsprozess

#### Markerdesign und Extraktionsgenauigkeit

Die Problematik des Markerdesigns und die Entwicklung entsprechender Bildoperatoren wurde im Bereich der Photogrammetrie und der Bildverarbeitung intensiv erforscht. Die Marker müssen hoch präzise extrahierbar sein, da die Qualität der Endergebnisse zum Grossteil von der Güte der Bildlokalisierung abhängt.
Die üblichen Marker liegen in Quadrat-, Kreis-, oder Punktform vor [92, 49, 83]. Brand [11] vergleicht die Lokalisierung von Punkten mit der Lokalisierung von Ecken quadratischer Marker und berichtet über eine Genauigkeit von jeweils 1/10 und 1/20 Pixel[1]. Außerdem wird gezeigt, dass sich die Punktlokalisierung bezüglich Rauschen weniger empfindlich als die Lokalisierung von Ecken verhält. Über eine extreme hohe Genauigkeit mit Punkten als Markern wird in [8] berichtet. In seinem Buch stellt H. Beyer eine photogrammetrische Kalibrierungsumgebung mit retro-reflektierenden Punkten vor und beschreibt Experimente, bei denen eine Präzision von 1/100 Pixel erreicht wurde. Um diese Genauigkeit zu erreichen, wurde von H. Beyer der Abbildungsprozess auf dem CCD-Chip der Kamera, sowie die Digitalisierung der Videokarte (Framegrabber) genau untersucht. Für kreisförmige Marker kann auch eine sehr gute Genauigkeit erreicht werden, sobald die perspektivische Verzerrung der Kreise im Bild berücksichtigt wird. Ergebnisse mit einer Präzision von 1/10 bis 1/20 Pixel werden in [83, 49] vorgestellt.
Aufgrund der in der Literatur vorgestellten Ergebnisse wurde im Rahmen dieser Arbeit ein Kalibrierungsobjekt mit weißen Punkten auf schwarzem Grund ausgewählt. Die Punkte besitzen einen Durchmesser von 1 cm und sind in einem $N \times M$ Gittern angeordnet. Um einerseits den Einfluss des Rauschens und die Lokalisierungsfehler zu minimieren und anderseits den gesamten Bildbereich gleichmäßig zu überdecken, wurde eine möglichst hohe Punktanzahl bevorzugt. Eine Aufnahme vom angewendeten Kalibrierungsobjekt ist in Abbildung 4.5 zu sehen.

#### Markerlokalisierung mit dem Tophat-Operator

Die Lokalisierung der Punkte soll mölichst automatisch erfolgen, da die Anzahl der Kalibrierungsmarker hoch ist und mehrere Aufnahmen für die Kalibrierung einer Kamera

[1] RMS zwischen reprojizierten 3D-Punkten und deren 2D-Lokalisierungen im Bild

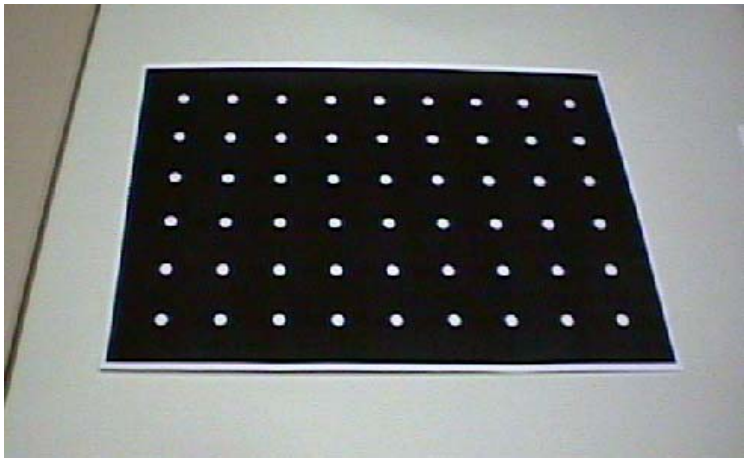
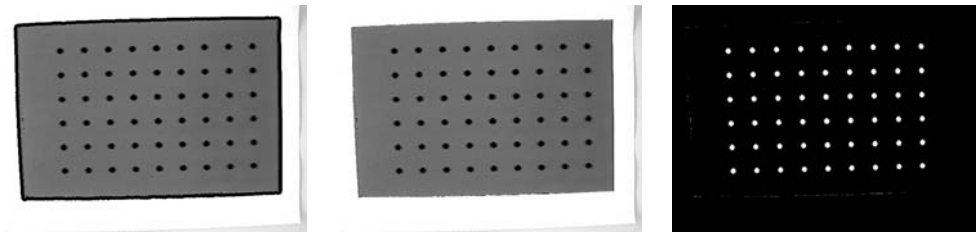


Abbildung 4.2: Kalibrierungsobjekt

benutzt werden. Um eine hohe Robustheit der Detektion zu erreichen, werden morphologische Bildoperatoren angewendet. Mit dem sogenannten Tophat-Operator kann gezielt nach Bildpunkten, d.h. kleinen, hellen Spots, gesucht und damit die Kalibrierungsmarker lokalisiert werden. In den folgenden Abschnitten wird der Tophat-Operator erläutert und mehrere Ergebnisse der Markerlokalisierung betrachtet. Auf weitere morphologische Operatoren wird im Rahmen dieser Arbeit nicht weiter eingegangen.

**Der Tophat-Operator:** Der sogenannte Tophat-Operator besteht genau genommen aus zwei Operationen:

- Im ersten Schritt wird auf das Bild ein sogenannter *opening*-Operator angewendet. Dieser Operator besteht aus einem *erosion*- gefolgt von einer *dilatation*-Operator.
- Im zweiten Schritt wird das Ergebnisbild vom Original subtrahiert.



(a) (b) (c)

Abbildung 4.3: Der Tophat-Operator: (a) *erosion* (b) *dilatation* (c) Subtraktion

Die Schritte des Tophat-Operators sind in Abbildung 4.3 veranschaulicht. Durch die *erosion* werden kleine, helle Regionen, d.h. in diesem Fall die gesuchten Punkte, entfernt, siehe Abbildung 4.3(b). Die *dilatation* kompensiert die Auswirkung der *erosion* wobei der *dilation*-Operator kein Einfluss auf die Region, die während der *erosion* zugeschlossen wurden, hat. Das resultierende Bild, in Abbildung 4.3(d) veranschaulicht, wird vom Originalbild subtrahiert. Die gesuchten Punkte sind damit gut lokalisiert und stellen wie in

Abbildung 4.3(c) gezeigt, helle Punkte auf einem vollständigen schwarzen Hintergrund dar.
Mit Hilfe einer einfachen Binarisierung und anschließend einer Berechnung der Schwerpunkte der weißen Regionen, können die Bildkoordinaten jedes einzelnen Punktes berechnet werden. Nach der Lokalisierung im Bild werden die Punkte möglichst präzise im Bild bestimmt. Hierfür wird ein zusätzlicher Operator (Extraktionsoperator) angewendet.

### Extraktion

Die Auswahl des Extraktionsoperators wird in erster Linie durch die folgenden Hauptkriterien (1) die Genauigkeit, (2) die Robustheit gegen Rauschen und (3) die Robustheit gegenüber ungleichmäßiger Beleuchtung bestimmt.
In [79] werden die Leistungen der sechs folgenden Operatoren verglichen:

1. Konturschwerpunkt
2. Binär-Markerschwerpunkt
3. Grauwertgewichteter Schwerpunkt
4. Quadratisch-grauwertgewichteter Schwerpunkt
5. Kontur-Fitting auf Basis einer Ellipse
6. Modellierung durch eine Gaußverteilung

Die Tests, die in [79] ausgeführt wurden, zeigen, dass ähnlich gute Ergebnisse mit der Berechnung des Markerzentrums gewichtet mit dem Quadrat der Pixelgrauwerte wie mit der Modellierung durch eine Gaußverteilung erreicht wurden. Im Rahmen dieser Arbeit wurde dieser erste Operator, “Grauwertgewichteter Schwerpunkt” auf Grund seiner guten Leistungen und seiner einfachen Implementierung für die Untersuchungen ausgewählt.

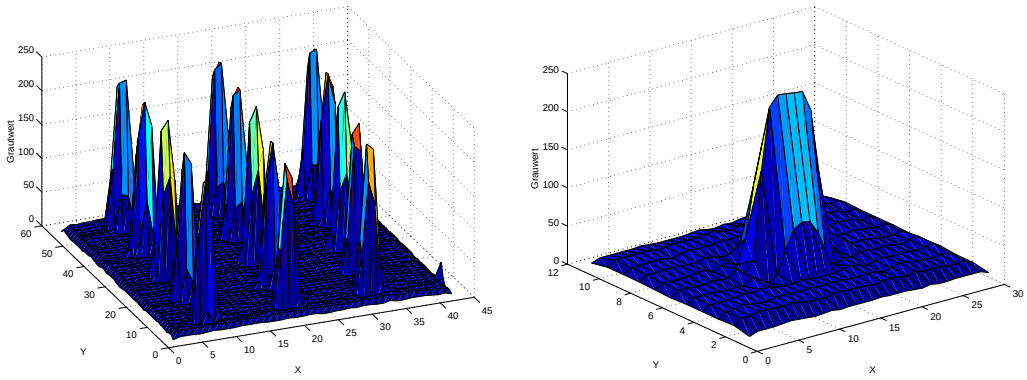



Abbildung 4.4: 3D-Darstellung der Marker

Der Operator kann wie folgt geschrieben werden:

$$\bar{x} = \frac{\sum_{j=1}^{h}\sum_{i=1}^{w} i.I_{ij}^2}{\sum_{j=1}^{h}\sum_{i=1}^{w} I_{ij}^2} \bar{y} = \frac{\sum_{j=1}^{h}\sum_{i=1}^{w} j.I_{ij}^2}{\sum_{j=1}^{h}\sum_{i=1}^{w} I_{ij}^2}$$

wobei $\bar{x}, \bar{y}$ die Bildkoordinaten der gesuchten Punkte, $I_{ij}$ den Pixelgrauwert an Bildposition $(i, j)$ und $(w, h)$ die Fensterdimensionen des Operators darstellen.

### 4.2.3 Kalibrierungsalgorithmus

#### Kalibrierung auf Basis einer Homographie

Die Beziehung zwischen einem 3D-Punkt $\boldsymbol{M}(X, Y, Z, 1)$ und seiner Projektion im Bild $\boldsymbol{m} = (u, v, 1)$ ist gegeben durch, siehe Kapitel 3:

$$s\boldsymbol{m} = \boldsymbol{A}(\boldsymbol{R}, \boldsymbol{t})\boldsymbol{M} \tag{4.1}$$

Das Kalibrierungsobjekt ist planar und es wird angenommen, dass es in der Ebene $Z = 0$ liegt. Die $i$-te Spalte der Rotationsmatrix $\boldsymbol{R}$ wird mit $\boldsymbol{r}_i$ bezeichnet. Daraus folgt:

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \boldsymbol{A}(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3)\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \boldsymbol{A}(\boldsymbol{r}_1, \boldsymbol{r}_2)\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{4.2}$$

$\boldsymbol{M'} = (X, Y, 1)^\top$ stellt den Punkt in der Ebene $Z = 0$ dar. Die Transformation zwischen dem Kalibrierungsobjekt und seiner Abbildung kann ebenfalls mit einer 2D-Homographie oder 2D-Kollineation $\boldsymbol{H}$ charakterisiert werden. Die Homographie $\boldsymbol{H}$ ist dann wie folgt definiert:

$$s\boldsymbol{m} = \boldsymbol{H}\boldsymbol{M'} \quad mit: \quad \boldsymbol{H} = \boldsymbol{A}(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{t}) \tag{4.3}$$

#### Berechnung der Homographie *H*

Es sei $\boldsymbol{X} = [\boldsymbol{h}_1^\top, \boldsymbol{h}_2^\top, \boldsymbol{h}_3^\top]^\top$, wobei $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, und $\boldsymbol{h}_3$ die Zeilen von $\boldsymbol{H}$ beschrieben. Für einen Punktepaar $\boldsymbol{m}_i, \boldsymbol{M}_i$) kann die Gleichung 5.7 folgendermaßen aufgestellt werden:

$$\begin{pmatrix} \boldsymbol{M}_i^\top & 0^\top & -u\boldsymbol{M}_i^\top \\ 0^\top & \boldsymbol{M}_i^\top & -v\boldsymbol{M}_i^\top \end{pmatrix}\boldsymbol{X} = 0 \tag{4.4}$$

Aus $n$ Punkten im Bild ergeben sich $2 \times n$ Gleichungen. Um das Gleichungssystem lösen zu können, werden von daher mindestens vier Punktpaare benötigt. Zur Lösung des Gleichungssystems wird ein SVD-Verfahren angewendet.

#### Numerische Stabilität

Eine numerische Instabilität wird dadurch verursacht, dass die Koordinaten der Punkte im Bild und in 3D bezüglich verschiedener Koordinatensysteme eingegeben worden sind und unterschiedliche Größe und Skalierung besitzen.
Solche Stabilitätsprobleme können durch eine Untersuchung der singulären Werte überprüft werden. Bei fehlerfreien Daten müssen mit Ausnahme des Wertes Null, der den Eigenvektor des Null-Raums definiert, alle Singularwerte gleich groß sein. Um das Stabilitätsverhalten zu untersuchen, werden zwei Stabilitätskoeffizienten $c_1$ und $c_2$ eingeführt. Für die Singularwerte $\sigma_1 >= \sigma_2 >= \ldots \sigma_9$ sind die Koeffizienten $c_1$ und $c_2$ folgendermaßen definiert:

$$\begin{aligned} c_1 = &\quad \sigma_8/\sigma_1 &\approx 1 \\ c_2 = &\quad \sigma_9/\sigma_8 &\approx 0 \end{aligned}$$

Eine Verbesserung der Stabilität ist erreicht, wenn die Daten in einem Intervall $[-1;1]$ vor der Berechnung des Gleichungssystems transformiert werden, siehe beispielsweise [46]. Nach der Berechnung von $\boldsymbol{H}$ werden die intrinsischen und externen Parameter der Kamera mit Hilfe der Gleichung 5.7 bestimmt.

#### Intrinsische Kameraparameter

$\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3)$ sei die im Abschnitt 4.2.3 definierte Homographie. Aus Gleichung 5.7 folgt:

$$(\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3) = s\boldsymbol{A}(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{t}) \tag{4.5}$$

Der Skalar $s$ wird durch die Vektoren $\boldsymbol{r}_1$ und $\boldsymbol{r}_2$ und den Verschiebungsvektor $\boldsymbol{t}$ beschrieben. $\boldsymbol{r}_1$ und $\boldsymbol{r}_2$ sind Vektoren einer Rotationsmatrix, d.h. sie stehen senkrecht zueinander und ihr Produkt ist gleich Null. Aus dieser Eigenschaft können zwei Gleichungen über die Matrix $\boldsymbol{A}$ abgeleitet werden:

$$\boldsymbol{h}_1^\top \boldsymbol{A}^{-\top} \boldsymbol{A}^{-1} \boldsymbol{h}_2 = 0 \tag{4.6}$$

$$\boldsymbol{h}_1^\top \boldsymbol{A}^{-\top} \boldsymbol{A}^{-1} \boldsymbol{h}_1 = \boldsymbol{h}_2^\top \boldsymbol{A}^{-\top} \boldsymbol{A}^{-1} \boldsymbol{h}_2 \tag{4.7}$$

Die Matrix $\boldsymbol{B} = \boldsymbol{A}^{-\top}\boldsymbol{A}^{-1}$ ist eine $3 \times 3$ symmetrische Matrix, d.h. $\boldsymbol{B}$ wird durch sechs Elemente beschrieben. Da eine Homographie $\boldsymbol{H}$ die zwei obigen Gleichungen liefert, werden drei Homographien, d.h. drei Bilder benötigt, um das System linear lösen zu können. Aus der Matrix $\boldsymbol{B}$ werden die einzelnen Parameter der Matrix $\boldsymbol{A}$ abgeleitet.

#### Externe Kameraparameter

Nach der Bestimmung der intrinsischen Parameter $\boldsymbol{A}$ werden die externen Kameraparameter aus der Matrix $\boldsymbol{H}$ abgeleitet. Die Gleichung 5.7 wird folgendermaßen geschrieben:

$$\boldsymbol{H}' = \boldsymbol{A}^{-1}\boldsymbol{H} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{t}) \tag{4.8}$$

Dem entspricht:

$$\begin{pmatrix} \boldsymbol{h}'_1 & \boldsymbol{h}'_2 & \boldsymbol{h}'_3 \end{pmatrix} = s \begin{pmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t} \end{pmatrix} \tag{4.9}$$

mit: $\boldsymbol{r}_1 = s\boldsymbol{h}_1$, $\boldsymbol{r}_2 = s\boldsymbol{h}_2$ und $t = s\boldsymbol{h}_3$ mit $s = 1/\|\boldsymbol{h}_1\| = 1/\|\boldsymbol{h}_2\|$ und $\boldsymbol{r}_3 = \boldsymbol{r}_1 \wedge \boldsymbol{r}_2$.
Die berechnete Matrix $\boldsymbol{R}$ hat auf Grund fehlerhafter Eingabedaten nicht die Eigenschaften einer Rotationsmatrix. Die Matrixvektoren $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3)$ betragen nicht die Länge eins und stehen nicht orthogonal zueinander. Die beste Annäherung von $\boldsymbol{R}$ mit einer Rotationsmatrix[2] wird mit Hilfe einer SVD-Zerlegung bestimmt [37].

### 4.2.4 Bundle Adjustment

Der im Abschnitt 4.2.3 präsentierte Algorithmus setzt voraus, dass die Kamera mit einer idealen Lochkamera übereinstimmt. In der Praxis ist diese Annahme jedoch oft nicht erfüllt, da viele Kameras starke, optische Verzerrungen erzeugen. Außerdem wird vorausgesetzt, dass die 3D-Koordinaten der Punkte des Kalibrierungsobjektes mit einer hohen

[2] nach dem minimalen, quadratischen Fehler und der Frobenius-Norm

Genauigkeit erfasst werden. Da das Kalibrierungsobjekt mit herkömmlichen Druckern auf einer deformierbare Auflage erstellt wird, stimmen die 3D-Vermessungen nicht überein. In diesem Abschnitt werden zuerst zusätzliche Parameter zur Modellierung der optischen Verzerrungen eingeführt. Anschließend wird eine Optimierungsmethode, die sogenannte "Bundle-Adjustment-Methode", präsentiert, die die Berechnung dieser Parameter und eine Verbesserung der Kalibrierungsgenauigkeit auf Basis einer Verfeinerung aller Daten, d.h. der Kameraparameter und der 3D-Koordinaten der Punkte $\boldsymbol{M}_i$, ermöglicht.

### Radiale Verzerrungen

Die radialen Verzerrungen werden mit Hilfe der zwei Parameter $k_1$ und $k_2$ modelliert. Die Gleichung zwischen den Punkten im Bild mit Koordinaten $\tilde{m}(\tilde{u}, \tilde{v})$ und den Punkten ohne Verzerrung $(u, v)$ ist folgendermaßen definiert:

$$\tilde{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{4.10}$$

$$\tilde{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{4.11}$$

wobei $x = u - u_0$, $y = v - v_0$, und $u_0$ und $v_0$ die Koordinaten des optischen Bildzentrums sind.

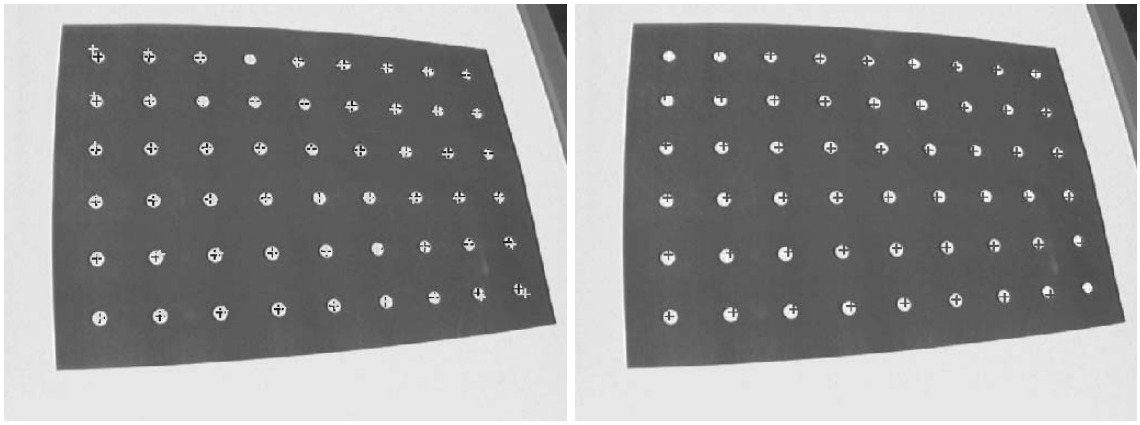


Abbildung 4.5: Berechnung der Bildverzerrung (Die weißen Kreuze stellen korrigierte Punkte dar)

### Bundle Adjustement

Die Parameter $k_1$ und $k_2$ werden durch ein Optimierungsverfahren oder "Bundle Adjustement" bestimmt. Das Verfahren approximiert das Minimum der Fehlerfunktion $\boldsymbol{F}(k_1, k_2)$, die den Fehler zwischen den detektierten $m_{ij}$ Bildpunkten und den reprojizierten Modellpunkten $\tilde{m_{ij}}$ berechnet. Die Funktion $\boldsymbol{F}$ ist wie folgt definiert:

$$\boldsymbol{F}(k_1, k_2) = \sum_{i=1}^{n} \sum_{j=1}^{m} \parallel m_{ij} - \tilde{m_{ij}}(k_1, k_2) \parallel^2 \tag{4.12}$$

wobei $m_{ij}$ den detektierten Punkt $i$ im Bild $j$ und $\tilde{m_{ij}}$ den reprojizierten 3D-Punkt $M_i$ im Bild $j$ darstellen.

Die mit $k_1$ und $k_2$ korrigierten Punkte sind in Abbildung 4.5 mit einem weißen Kreuz veranschaulicht. Für eine herkömmliche Kamera mit Weitwinkel sind die Fehler, die durch die Linsenverzerrung verursacht sind, sehr hoch und können, wie Tabelle 4.2 und Abbildung 4.5 zu entnehmen ist, bis ca. 10 Pixels betragen.
Nach der Berechnung der Parameter $k_1$ und $k_2$ wird eine globale Optimierung über aller Parameter der Kamerakalibrierung vorgenommen. Dabei werden sowohl die Kameraparameter als auch die 3D-Koordinaten der Punkte des Kalibrierungsobjektes optimiert. Die Fehlerfunktion $\boldsymbol{F}$ ist in diesem Fall folgendermaßen definiert:

$$\boldsymbol{F}(A, k_1, k_2, R_j, t_j, M_i) = \sum_{i=1}^{n} \sum_{j=1}^{m} \parallel m_{ij} - \tilde{m_{ij}}(A, k_1, k_2, R_j, t_j, M_i) \parallel^2 \tag{4.13}$$

Die Kalibrierungsfehler können mit dem Bundle-Adjustement-Verfahren stark verringert werden. Ergebnisse der Optimierung werden im nächsten Abschnitt vorgestellt.

### 4.2.5 Ergebnisse der Kamerakalibrierung

Die Genauigkeit der Kamerakalibrierung ist von der Qualität der Bilder stark abhängig, da ein großer Teil der Fehler auf die Lokalisierung im Bild zurückzuführen ist. Die Bilder müssen möglichst mit einem Stativ und einer diffusen, gleichmäßigen Beleuchtung aufgenommen werden.
Durch Tabelle 4.2 und 4.1 werden Kalibrierungsergebnisse zweier Kameras vorgestellt. Die verwendete Toshiba- Kamera ist eine hochwertige Kamera, die kaum Linsenverzerrung vorweist. Die Kalibrierungsfehler liegen bei ca. 1.2 Pixel nach Berechnung der Kameraparameter mit dem linearen Lösungsverfaren, siehe Abschnitt 4.2.3 und nur 0.06 Pixel nach Berechnung der Parameter $k_1$ mit Durchführung des Bundle-Adjustments.

| | Ohne Optimierung | | Mit Optimierung | | |
|---|---|---|---|---|---|
| | error mean | error max | error mean | error max | $k_1$ |
| 6 images | 1.22 | 4.30 | 0.068 | 0.27 | -1.7E-7 |
| 5 images | 1.23 | 4.48 | 0.060 | 0.21 | -1.9E-7 |
| 4 images | 1.22 | 4.16 | 0.042 | 0.15 | -1.8E-7 |

Tabelle 4.1: Toshiba-Kamera

| | Ohne Optimierung | | Mit Optimierung | | |
|---|---|---|---|---|---|
| | error mean | error max | error mean | error max | $k_1$ |
| 6 images | 2.54 | 10.7 | 0.17 | 0.55 | -7.5E-7 |
| 5 images | 2.58 | 9.26 | 0.16 | 0.53 | -7.7E-7 |
| 4 images | 2.20 | 8.41 | 0.15 | 0.39 | -8.9E-7 |

Tabelle 4.2: Pyros Kamera

Die Pyros-Kamera ist eine vergleichsweise einfache Kamera mit einem Weitwinkelobjektiv, das Plastiklinsen beinhaltet. Diese Kamera kann nur annährend mit dem idealen Lochkameramodell nachgebildet werden. Aus diesem Grund fallen die Kalibrierungsfehler

entsprechend hoch aus, d.h. 2.5 Pixel im Durchschnitt und bis 10 Pixel für die Kalibrierungspunkte am Rand des Bildes. Durch die Optimierung und die Modellierung der Linsenverzerrung können diese Ergebnisse stark verbessert werden. Die durchschnittlichen Fehler betragen ca. 1/6 Pixel, wobei der maximale Fehler auf 1/2 Pixel reduziert werden konnte.
Die Ergebnisse der Kamerakalibrierung sind insgesamt nach dem Optimierungsschritt sehr genau. Diese Untersuchungen zeigen, dass die Kameraparameter auch mit einem einfachen Kalibrierungsobjekt präzis nachgebildet werden können, jedoch unter der Voraussetzung, dass (1) die Verzerrung der Linsen berücksichtigt wird und dass (2) die 3D-Koordinaten der Kalibrierungspunkte korrigiert werden.

## 4.3 Bildberarbeitung auf Basis einer Ansicht

In diesem Abschnitt wird die Erweiterung einer realen Szene mit virtuellen Objekten auf Basis einer einzigen Szeneansicht betrachtet. Die Berechnung der Kameraparameter aus einer Sicht ist im Bereich Computer-Vision ein intensiv erforschtes Thema . Deswegen wird ein Überblick über die relevanten Algorithmen gegeben und anschließend ein Konzept zu einer praktischen Implementierung erarbeitet und umgesetzt.

### 4.3.1 Gewinnung aller Kameraparameter

#### Die Algorithmen

Wie im Kapitel 3 dargelegt wurde, können die Kameraparameter aus einem gegebenen Bild nach dem DLT-, Tsai- oder Weng- Verfahren berechnet werden [85, 90]. Die Rückrechnung der Kameraparameter aus einem gegebenen Bild ist jedoch nummerisch nicht stabil.
Um dennoch gute Ergebnisse zu erzielen, müssen mindestens 15 bis 20 Passpunkte, die möglichst gleichmäßig über das Bild verteilt sind, präzise eingegeben werden. In der Praxis stehen meist zu wenige Punkte zur Verfügung, so dass auf dieser Weise keine korrekten Ergebnisse erzielt werden können.
Wenn alle Punkte in einer Ebene liegen, kann das Algorithmus wie im Abschnitt 4.2 vorgestellt, angewendet werden. Dennoch aus einer Sicht kann nur die Brennweite, das Aspect-Ratio abgeleitet werden.

#### Eingabe der Kameraparameter in das Rendering-System

Ein weiteres Problem stellt die Eingabe der Kameraparameter in das Rendering- System dar. Einige Systeme, die ursprünglich für reine Graphikapplikationen konzipiert wurden, lassen nur wenige Parameter, wie Position, Orientierung und Field of View zu. Der Aspect-Ratio oder der Mittelpunkt der Bildebene der Kamera können nicht berücksichtigt werden. Die Vernachlässigung dieser Parameter führt zu einem ungenaueren Ergebnis. Für mit OpenGL geschriebene Software, wie z.B. OpenSG [66], gibt es die Möglichkeit direkt das OpenGL-Frustum oder sogar die Projektionsmatrix $\boldsymbol{P}$ in das System einzugeben, [88]). Eine direkte Übernahme der Projektionsmatrix ist von Vorteil, da allein die Faktorisierung von $\boldsymbol{P}$ in die intrinsischen und externen Kameraparameter einen Genauigkeitsverlust verursacht.

Um eine bessere Stabilität zu erzielen, erweist es sich als günstig die Kamera vorher zu kalibrieren und für eine gegebenes Bild nur die externen Kameraparameter neu zu berechnen.

### 4.3.2 Gewinnung der externen Kameraparameter

Bei der Verwendung kalibrierter Kameras müssen für das zu bearbeitenden Bild nur noch die jeweilige Position und Orientierung der Kamera zurückgewonnen werden. Dieses Problem wird im Bereich der Photogrammetrie als die Bestimmung der *absoluten Orientierung* bezeichnet. Die absolute Orientierung wird durch insgesamt sechs Parameter (3 Rotationswinkel und 3 Positionskoordinaten) bestimmt. Im folgenden wird ein Überblick über mögliche Lösungsansätze zur Bestimmung der absoluten Orientierung aus vorliegenden Bildern gegeben.

#### Lösungen mit drei Punkten

Um die maßgeblichen sechs Parameter berechnen zu können, sind als minimale Eingabemenge drei Punkte notwendig.

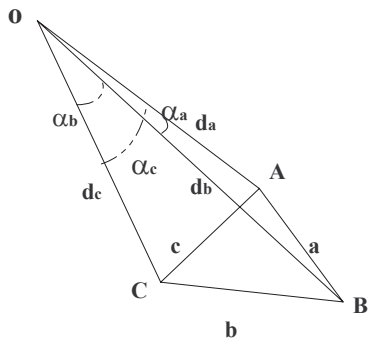



Abbildung 4.6: Geometrie der Kameraposition und -orientierung und der 3D-Szenenpunkten

Zur Bestimmung der Kameraposition lassen sich für die Punkte $\boldsymbol{A}, \boldsymbol{B}$ und $\boldsymbol{C}$ mit den jeweiligen Abständen $d_a, d_b$ und $d_c$ zum Kamerazentrum $\boldsymbol{o}$ und die Abstände $a$, $b$ und $c$ zwischen den Punkten $\boldsymbol{A}, \boldsymbol{B}$ und $\boldsymbol{C}$, die folgenden drei Gleichungen formulieren:

$$a^2 = d_a^2 + d_b^2 - 2d_a d_b \cos(\alpha_a) \tag{4.14}$$

$$b^2 = d_b^2 + d_b^2 - 2d_b d_c \cos(\alpha_b) \tag{4.15}$$

$$c^2 = d_c^2 + d_a^2 - 2d_c d_a \cos(\alpha_c) \tag{4.16}$$

wobei $\alpha_a$, $\alpha_b$ und $\alpha_c$ die Winkel zwischen den jeweiligen Strahlen darstellen, siehe Abbildung 4.6.
Die resultierenden Gleichungen sind nichtlinear, können aber analytisch gelöst werden. In der Literatur werden zahlreiche Lösungen der drei Bestimmungsgleichungen vorgeschlagen, die in [45] zusammengefasst und verglichen wurden. Weitere Lösungsmethoden können in [44, 45, 35] nachgelesen werden. Es ist jedoch zu berücksichtigen, dass die beschriebenen Drei-Punkte-Verfahren keine eindeutige, sondern sechs mögliche Lösungen liefern.

Als Eingabedaten können auch Linien verwendet werden. In [21] wird ein linearer Algorithmus vorgestellt. Auch in diesem Fall liefert eine Drei-Linien-Lösung kein eindeutiges Ergebnis, sondern mehrere möglichen Lösungen.
Andere Ansätze basieren auf iterativen oder nicht-linearen Verfahren [58, 68, 15], bei denen Initialwerte eingegeben werden müssen. Die Lösung von Phong und Horaud [68] wurde im System *CamCal* auf Basis des Levenbergt-Marquardt-Optimierungsverfahrens [69] implementiert. Bei diesem System muss eine erste Lösung für die Kameraposition und -orientierung interaktiv vom Benutzer eingeben werden.
Problematisch erweist sich bei Drei-Punkte-Lösungen die direkte Abhängigkeit der Berechnung der absoluten Orientierung von der Genauigkeit der Eingabedaten. Für das Ausgangsbild, auf dessen Basis die Minimierung durchgeführt wird, existiert immer eine exakte Lösung. Dennoch kann keine Aussage über die Korrektheit der Kameraorientierung getroffen werden, und das Zusammenwirken von Ungenauigkeiten der 2D- und 3D-Eingabe führt meist zu falschen Überlagerungen. Insbesondere werden, wenn das virtuelle Objekt von den Passpunkten weiter entfernt liegt, starke Missregistrierung des virtuellen Objektes durch kleine Fehler der Eingabedaten verursacht.

#### Lösung mit vier und mehr Punkten

Für vier und fünf Punkte existieren bislang nur wenige Verfahren, die eine direkte und eindeutige Lösung liefern. Entsprechende Algorithmen wurden erst kürzlich in [70, 84] veröffentlicht. Für sechs und mehr Punkte kann das im Kapitel 3 vorgestellte DLT-Verfahren angewendet werden.
Iterative Lösungen existieren schon seit längerer Zeit und sind beispielsweise in [20, 44] zu finden. Diese Ansätze haben oft den Vorteil, robust zu sein, und sind für die Praxis besonders relevant.
Entsprechend der Kamerakalibrierung kann die spezielle Konfiguration, für die alle Punkte auf einer Ebene liegen, getrennt betrachtet werden. Wie im Abschnitt 4.2 vorgestellt, werden aus der Homographie zwischen den Raum- und Bild-Punkten die Rotationsmatrix und der Translationsvektor abgeleitet. Andere Methoden, mit denen speziell Vierecke ausgewertet werden, können beispielsweise [52, 43, 2] entnommen werden.

### 4.3.3 Untersuchungen und Vergleich der Vier-, Fünf- und Sechs-Punkte-Lösungen

Der Vergleich der Algorithmen basiert auf einer vordefinierten Kameraposition und -orientierung und einer festen 3D-Punktkonfiguration. Für die Untersuchungen werden die 3D-Punkte auf die Bildebene projiziert und bilden dadurch sogenannte *Ground-Truth*-Daten, für welche eine fehlerfreie Berechnung der Kameraposition und -orientierung möglich ist. Anschließend wird systematisch weißes Rauschen wechselnder Stärke zu den Bildpunkten addiert und das Verhalten, d.h. die Robustheit der implementierten Algorithmen analysiert.
Dafür wird zuerst die gefundene Rotationsmatrix $\boldsymbol{R}$ mit der vordefinierten Kameraorientierung, d.h. mit der *Ground-Truth*-Matrix $\boldsymbol{R}_g$, verglichen. Die Matrix $\boldsymbol{R}$ wird transponiert und mit $\boldsymbol{R}_g$ multipliziert. Das Ergebnis soll im Fall einer fehlerfreien Berechnung die Identitätsmatrix bilden, für welche die Frobenius-Norm gleich eins ist. Je mehr $\boldsymbol{R}$ von dem richtigen Matrix $\boldsymbol{R}_g$ abweicht, desto mehr wächst der Normbetrag der Frobeniusnorm. Für

die Fehlerschätzung des Positionsvektors $\boldsymbol{T}$ wird einfach der Abstand mit dem *Ground-Truth*-Positionvektor $\boldsymbol{T}_g$ für die verschiedenen Rauschpegel gebildet. Den letzten Indikator für die Robustheit der Algorithmen liefert der sogenannte Reprojektionsfehler $\boldsymbol{E}_p$ der 3D-Punkte in den Bildern. Hierfür wird der Mittelwert über alle Punkte berechnet und als Kriterium angesetzt.
Jedes Experiment wird für jeden Rauschpegel 100 mal wiederholt und die Fehler von $\boldsymbol{R}$, $\boldsymbol{T}$ und $\boldsymbol{E}_p$ gemittelt.

#### Untersuchungen der ausgewählten Vier-Punkte-Lösungen

Für die Vier-Punkte-Lösungen wurden fünf verschiedene Algorithmen implementiert und verglichen, wobei zwei eine lineare und drei eine iterative Lösung liefern. Es handelt sich einerseits um die Algorithmen von Daniilidis [3]und Quan-Lan [70] und andererseits um die Algorithmen von Lu-Hager [59], Dementhon (*POSIT*) [20] und einer nichtlinearen Lösung (*NLLS*) auf Basis des Levenberg-Marquart-Verfahrens [69].
Die Abbildungen 4.7, 4.8 und 4.9 stellen die Ergebnisse der Versuche dar, wobei die Standardabweichung $\sigma$ des Rauschens einen Zahlenbereich von Null bis Zehn annimmt.
Die nicht-linearen Algorithmen verhalten sich zueinander ähnlich und weisen eine geringe Fehlergröße auf. Wie zu erwarten war, liefern sie sehr genaue Ergebnisse. Bemerkenswert ist die Stabilität der Lösung nach Daniilidis, die obwohl es sich um eine lineare Lösung handelt, Fehler der Größenordnung nichtlinearer Lösungen liefert und sogar zu etwas genaueren Ergebnissen als der *POSIT*-Algorithmus führt. Im Vergleich dazu verhält sich die Lösung nach Quan-Lan relativ rauschempfindlich.

#### Untersuchungen der ausgewählten Fünf-Punkte-Lösungen

Um die Kameraposition und -orientierung mit fünf vorgegebenen Punkten zu bestimmen, können dieselben Algorithmen, die im vorangehenden Abschnitt bezüglich der Vier-Punkt-Lösung beschrieben wurden, angewendet werden. Dabei können mit fünf Punkten ähnliche Ergebnisse bezüglich Stabilität und Genauigkeit der verschiedenen Lösungen können beobachtet werden, siehe Abbildung 4.10, 4.11 und 4.12.

#### Untersuchungen der ausgewählten Sechs-Punkte-Lösungen

Für sechs Punkte existieren zur Berechung der Kameraposition und -orientierung zusätzliche Algorithmen. Das Standard DLT-Verfahren, siehe Kapitel 3, auf Basis einer Systemlösung mit SVD und die Lösung nach Fiore [34] wurden im Rahmen der Arbeit zusätzlich implementiert und mit den vorherigen Algorithmen verglichen.
Wie die Abbildungen (4.13,4.14,4.15) zeigen, ist das DLT-Verfahren sehr empfindlich und liefert die schlechtesten Ergebnisse. Die Lösung nach Fiore liefert eine relativ fehlerbehaftete Berechnung von $\boldsymbol{R}$ und $\boldsymbol{T}$, wobei die Reprojektionsfehler im Bild die Größenordnung der genauesten Algorihmen (Lu-Hager, NLLS) vorweisen.

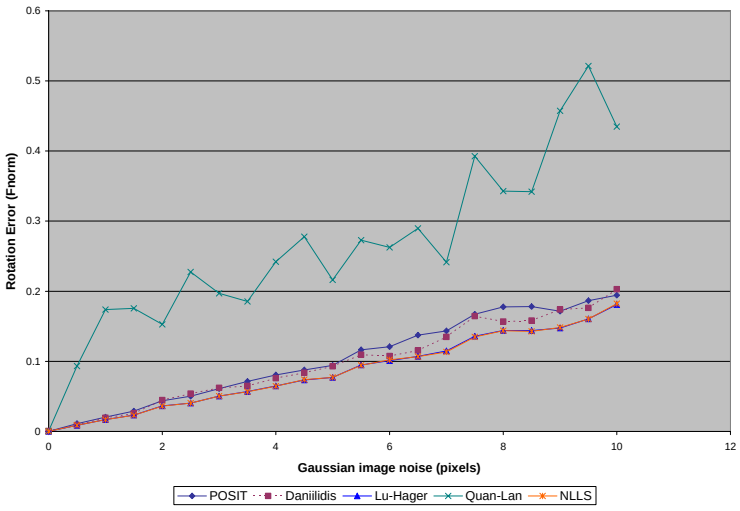



Abbildung 4.7: Fehler der Orientierungsbestimmung $\boldsymbol{R}$ (Vier-Punkte-Lösung)

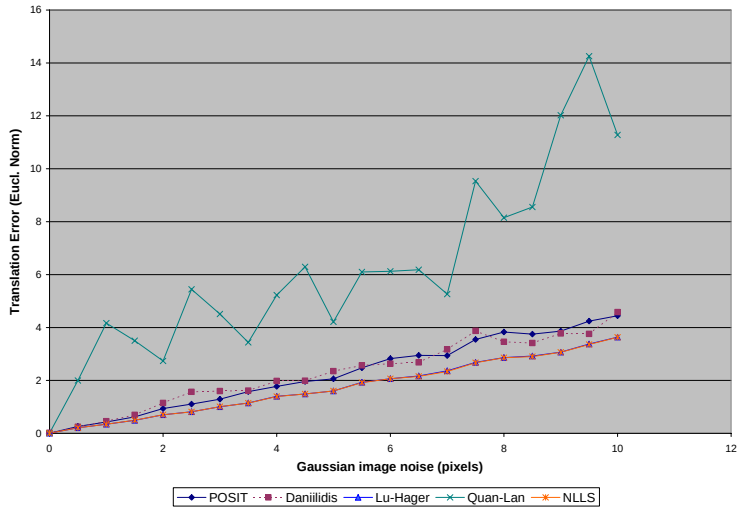



Abbildung 4.8: Fehler der Positionsbestimmung $\boldsymbol{T}$ (Vier-Punkte-Lösung)

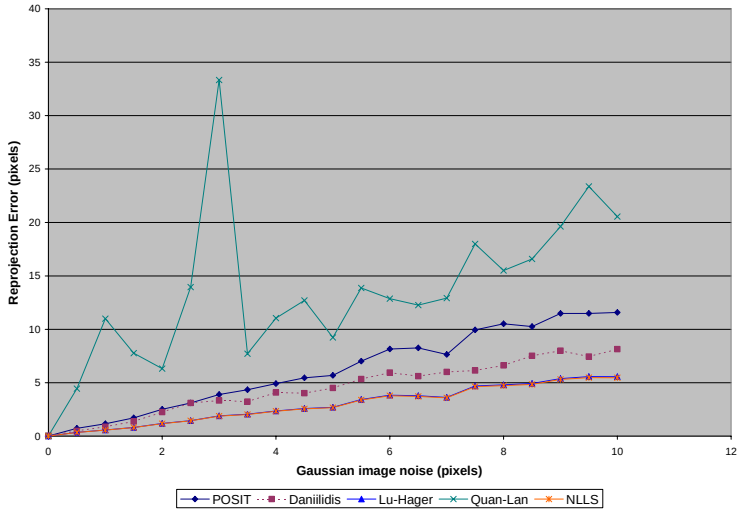



Abbildung 4.9: Reprojektionsfehler (Vier-Punkte-Lösung)

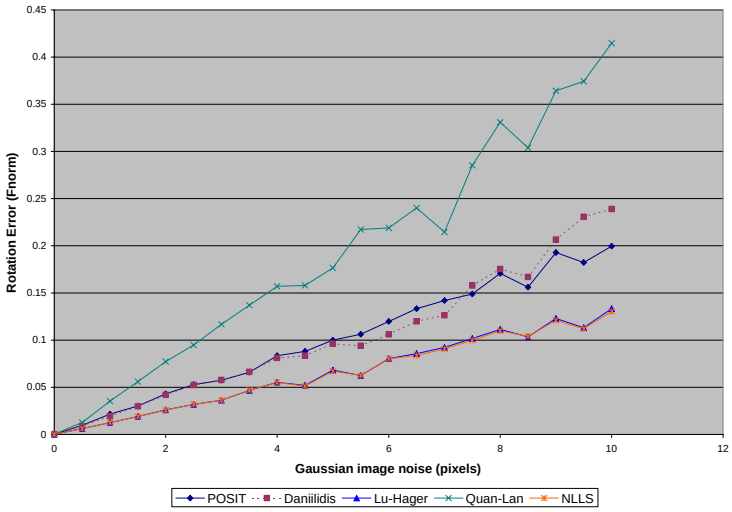



Abbildung 4.10: Fehler der Orientierungsbestimmung $\boldsymbol{R}$ (Fünf-Punkte-Lösung)

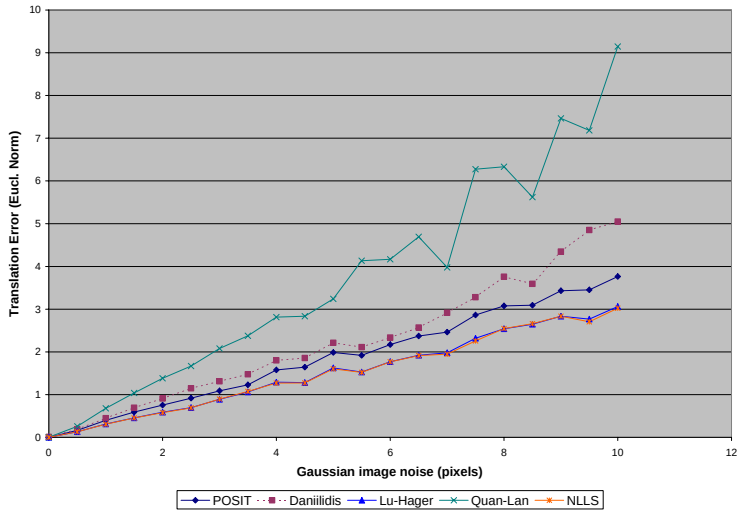



Abbildung 4.11: Fehler der Positionsbestimmung $\boldsymbol{T}$ (Fünf-Punkte-Lösung)

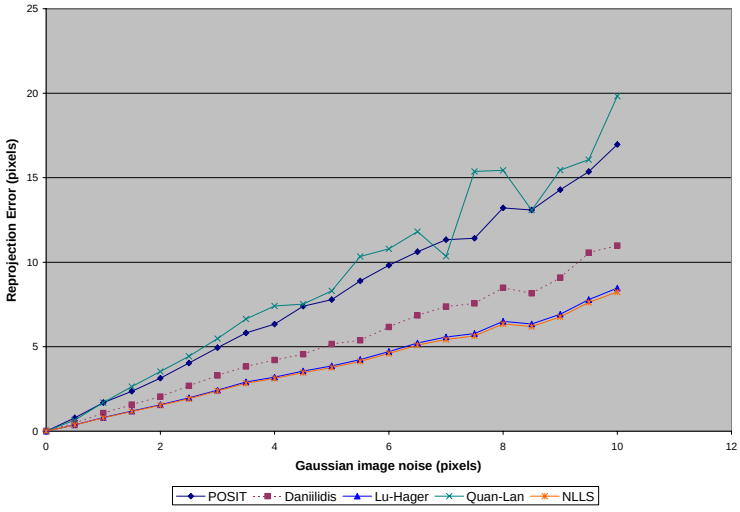



Abbildung 4.12: Reprojektionsfehler (Fünf-Punkte-Lösung)

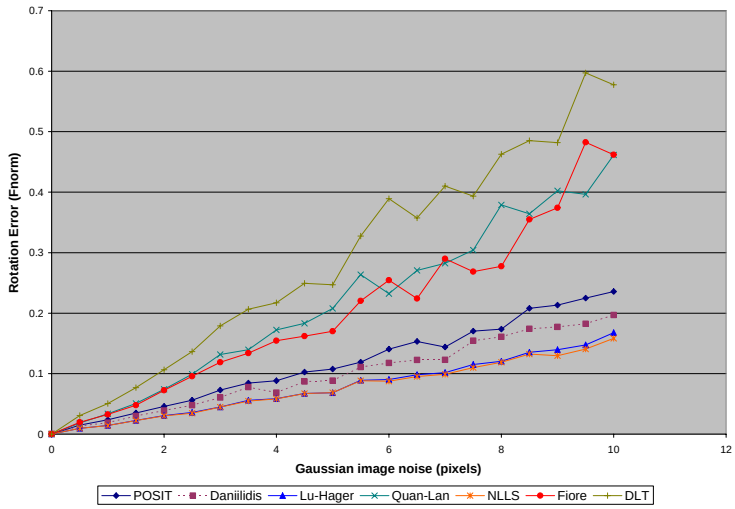



Abbildung 4.13: Fehler der Orientierungsbestimmung $\boldsymbol{R}$ (Sechs-Punkte-Lösung)

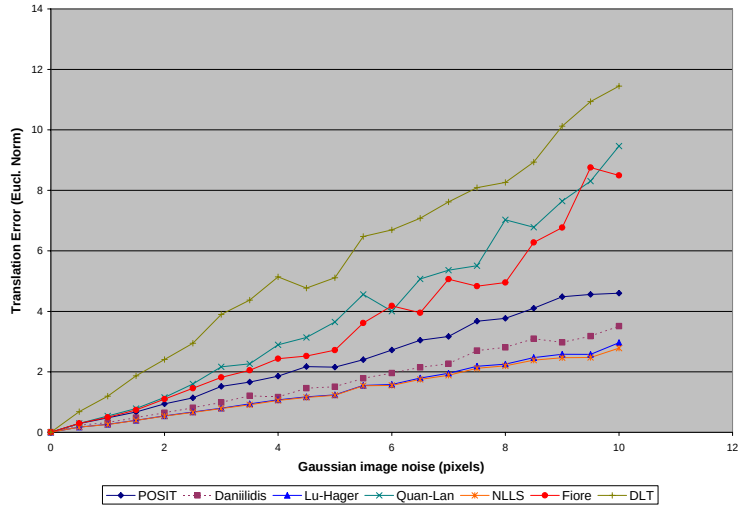



Abbildung 4.14: Fehler der Positionsbestimmung $\boldsymbol{T}$ (Sechs-Punkte-Lösung)

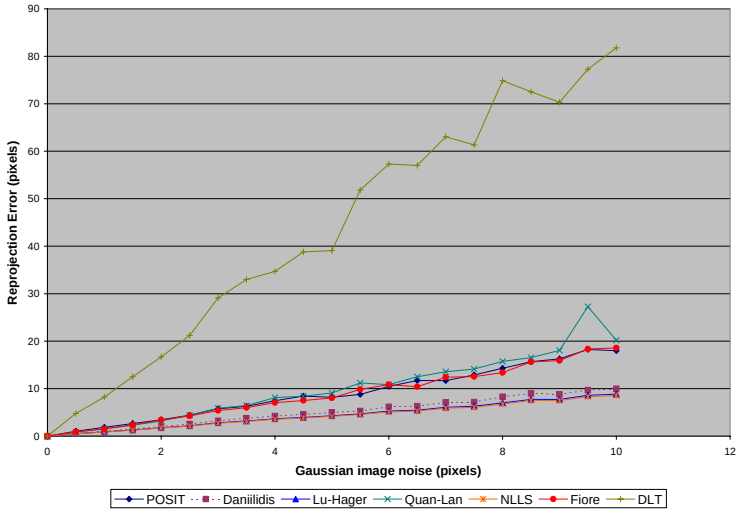



Abbildung 4.15: Reprojektionsfehler (Sechs-Punkte-Lösung)

### 4.3.4 Das CamCal-Tool: Kalibrierung anhand von Daten eines virtuellen Modells (CAD)

Das CamCal-Tool besitzt die Zielsetzung, die Erzeugung von AR-Bildern so leicht wie möglich zu gestalten, indem alle benötigten Funktionalitäten (Bildgrabbing, Punkteeingabe, Kalibrierung, und Rendering) in das Programm integriert werden. Im folgenden werden sowohl die Programmanwendung als auch die Algorithmen, auf denen dieses Tool basiert, erläutert und anschließend wesentliche Merkmale dieses Tools zusammengestellt.

#### Vorgehensweise

Im ersten Schritt werden aus einem Live-Video die relevanten Bilder ausgewählt und gespeichert. Um die Bilder anschließend zu kalibrieren, werden Passpunkte benötigt. Diese werden interaktiv in den einzelnen Bilder angeklickt und so als 2D-Punkte markiert. Um nun die Position des jeweiligen 2D-Punktes im Raum zu bestimmen, wird eine Socket-Verbindung mit dem CAD- oder VR-System aufgebaut. Für jeden 2D-Passpunkt kann daraufhin im VR-Modell der entsprechende 3D-Punkt selektiert werden.

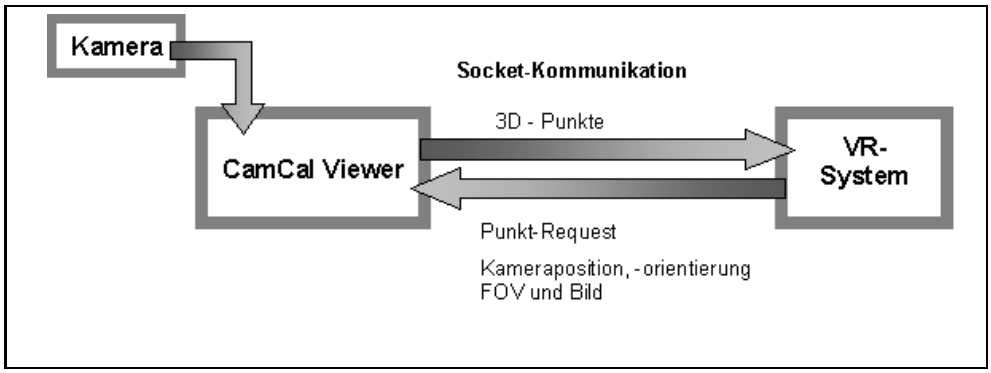



Abbildung 4.16: Systemstruktur

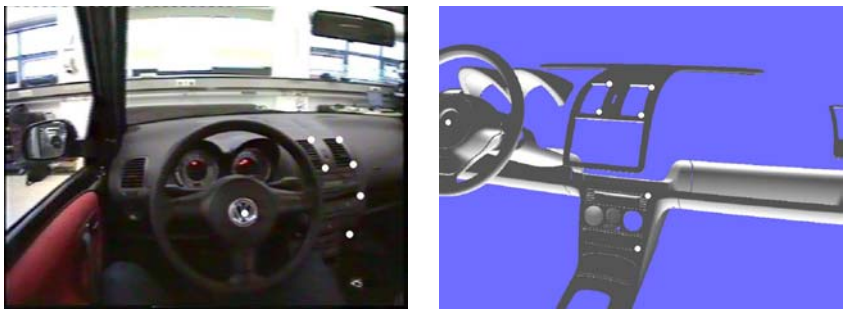


Abbildung 4.17: Video- and VR-viewers

Durch diese Vorgehensweise entstehen die für die Berechnung der absoluten Orientierung benötigten 2D/3D-Punktpaare. Nach der Berechnung der Kameraorientierung werden die einzelnen Parameter (Position, Rotation und Blickfeld) und das zu erweiternde Bild an das Renderingsystem weitergeleitet.
Die Bilder in Abbildung 4.18 zeigen ein Ergebnis, das mit Hilfe CamCal-Tools erzeugt wurde. Bild (a) zeigt das ursprüngliche Bild, im mittleren Bild wurde das Ausgangsbild

(a) (b) (c)

Abbildung 4.18: (a) Das ursprüngliche Bild; (b) Überlagerung mit dem VR-Modell; (c) Wireframe Modus

durch ein VR-Modell überlagert und Bild (c) auf der rechten Seite stellt die Überlagerung im Wireframe Modus dar.

#### Algorithmen zur Bestimmung der externen Kameraparameter

Um die Anzahl der Interaktionsschritte zu minimieren, sind Algorithmen notwendig, mit denen anhand möglichst weniger Punktinformationen die Kameraparameter bestimmt werden können. Da die Eingabe dreier Punkte zur keiner eindeutigen Lösung führt, sollten jedoch mindestens vier Punkte zur Verfügung stehen.
Zur Implementierung des CamCal-Tools wurden drei verschiedenen Algorithmen verwendet. Es handelt sich hierbei um den Dementhons-Algorithmus [20], das nichtlineares Optimierungsverfahren [68] und das DLT-Verfahren, das mindestens 6 Punkte. Je nach Konfiguration und Präzision der Eingabe muss der geeignetste Algorithmus ausgewählt werden. Daher werden zunächst alle Algorithmen systematisch aufgerufen, die jeweiligen Ergebnisse auf Basis der Fehler im Bild untersucht und die beste Lösung weiterverwendet. Diese Vorgehensweise hat sich als sehr robust erwiesen.

## 4.4 Bildverarbeitung auf Basis mehrerer Ansichten

### 4.4.1 Einleitung

Anhand mehrerer Ansichten einer Szene kann man, rein intuitiv betrachtet, besser die Position und Größe einzelner Objekte einschätzen. Die Problematik besteht darin, solche subjektiven Feststellungen mathematisch zu beschreiben und zu quantifizieren. Eine mathematische Formulierung ermöglicht, alle benötigten Kamera- und Szeneninformationen einzig aus den Bildern zu gewinnen, woraus eine flexible und leicht einsetzbare Vorgehensweise resultiert. Für diesen Zweck werden *structure and motion*-Techniken der Computer-Vision angewendet und spezifische Lösungen für die Bilderweiterung entwickelt.

### 4.4.2 Bearbeitungsprozess

Um aus mehreren Ansichten die Kameraparameter zu bestimmen, wird zunächst die relative Kameraorientierung aus der Zuordnung von Punkten bzw. Linien zwischen den Bildern berechnet. Anschließend kann ein einfaches 3D-Modell der Szene durch Triangulation er-

zeugt werden. Alle notwendigen Informationen zur Erweiterung eines Bildes mit virtuellen Objekten stehen dann zur Verfügung.
Die Schritte zum Bilderweiterung werden folgendermaßen zusammengefasst:

- Eingabe der Korrespondenzpunkte in $n$-Bilder
- Berechnung der Kameraparameter
- Konstruktion eines 3D-Modells
- Einfügen und Positionierung des virtuellen Objektes im 3D Raum
- Rendering der Augmented Images

In den folgenden Abschnitten wird vorausgesetzt, dass die intrinsischen Kameraparameter für die verschiedenen Ansichten bekannt sind, d.h. dass die Ansichten von kalibrierten Kameras aufgenommen wurden.

### 4.4.3 Kalibrierte Kameras

#### Berechnung der relativen Orientierung

#### Berechnung der $\boldsymbol{E}$-Matrix

Wie im Kapitel 3 erläutert, gilt für ein Punktpaar $\boldsymbol{m}$ im ersten Bild und $\boldsymbol{m'}$ im zweiten Bild die Gleichung:

$$\boldsymbol{m'}^{\top} \boldsymbol{E} \boldsymbol{m} = 0 \tag{4.17}$$

mit der essentiellen Matrix $\boldsymbol{E}$.
Die Matrix $\boldsymbol{E}$ wird mit Hilfe des Acht-Punkt-Algorithmus [57] berechnet, wobei eine SVD für die Lösung des Gleichungssystems angewendet wird, siehe Abschnitt 3.3.3. Die Eigenschaften einer essentiellen Matrix werden hierbei berücksichtigt und mit einer SVD-Dekomposition erzwungen.
Die relative Rotation $\boldsymbol{R}$ und Translation $\boldsymbol{t}$ zwischen zwei Kameras kann nach der im Abschnitt 3.4.3 beschriebenen Methode berechnet werden.

#### Planare Konfiguration

Planare Szenen treten in der Praxis relativ oft auf, beispielsweise in Innenräumen. Punkte auf einer Ebene im Raum werden in den Bildern über eine Homographie $\boldsymbol{H}$ transformiert. Für die Punkte $\boldsymbol{m}$ im ersten und $\boldsymbol{m'}$ im zweiten Bild gilt:

$$\boldsymbol{m'} = \boldsymbol{H} \boldsymbol{m} \tag{4.18}$$

Wie im Kapitel 3 vorgestellt, sind zwei Lösungen $\boldsymbol{R}$ und $\boldsymbol{t}$ möglich.

#### Twisted pair und Visibility Test

Die beiden Methoden für planare und 3D-Szene liefern zwei mögliche Lösungen für $\boldsymbol{R}$ und $\boldsymbol{t}$, die in der Literatur als *twisted pair* bezeichnet werden. Nur eine der beiden Lösungen ist jedoch physikalisch möglich. Bei dieser Lösung muss die Tiefe der eingegebenen Punkte

positiv sein, da die z-Achse der Kamera in Richtung der Szene modelliert ist. Aus diesem Grund wird dieser Überprüfung *visibility test* genannt.

Als Kriterium wird auf Grund von Ungenauigkeiten oder falscher Punktzuordnung, nicht die Tiefe einzelner Punkte sondern die Summe der Tiefen über alle Punkte gewählt.

Die Tiefe $z$ wird entsprechend der Gleichung 3.48 des im Kapitel 3 beschriebenen Verfahrens berechnet.

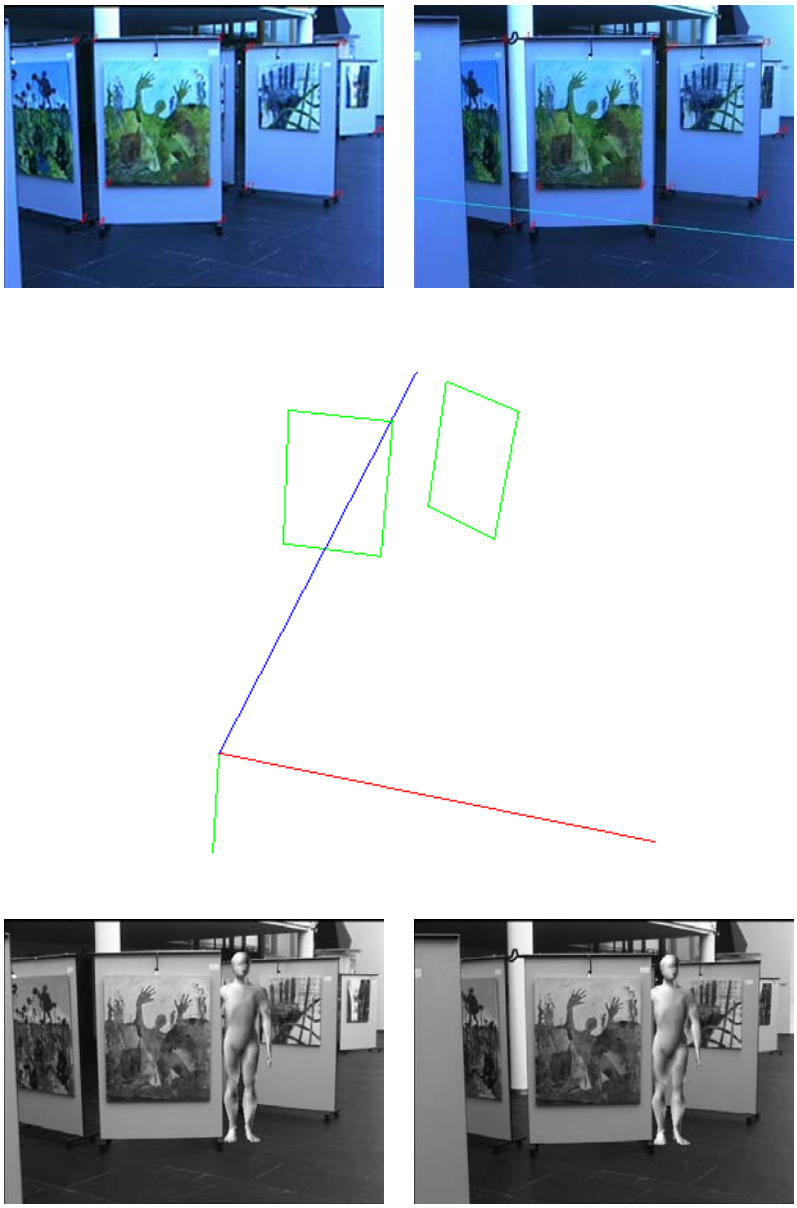


Abbildung 4.19: Bilderweiterung auf Basis der Berechnung der $\boldsymbol{E}$-Matrix

### Mehr als zwei Ansichten

Für zwei Ansichten wird die Norm des Translationsvektors auf 1 ($\|\boldsymbol{t}\| = 1$) gesetzt und dadurch eine beliebige Skalierung festgelegt. Für die dritte und folgende Ansicht muss die Länge von $\boldsymbol{t}$ abgeleitet werden.
$\boldsymbol{R}_{ij}, \boldsymbol{R}_{ik}, \boldsymbol{R}_{jk}$ und $\boldsymbol{t}_{ij}, \boldsymbol{t}_{ik}, \boldsymbol{t}_{jk}$ bilden die Rotationsmatrix bzw. Translationsvektoren zwischen den drei Ansichten $I$, $J$ und $K$, die paarweise mit dem oben beschriebenen Verfahren bestimmt wurden.
Die Projektionsmatrizen $\boldsymbol{P}_i$, $\boldsymbol{P}_j$ und $\boldsymbol{P}_k$ der drei Bilder werden folgendermaßen definiert:

$$\boldsymbol{P}_i = \boldsymbol{A}_i(\boldsymbol{I}, 0) \tag{4.19}$$
$$\boldsymbol{P}_j = \boldsymbol{A}_j(\boldsymbol{R}_{ij}, \boldsymbol{t}_{ij}) \tag{4.20}$$
$$\boldsymbol{P}_k = \boldsymbol{A}_k(\boldsymbol{R}_{ik}, \gamma_k \boldsymbol{t}_{ik}) \tag{4.21}$$

$\gamma_k$ stellt hier den gesuchten Skalierungsfaktor zwischen den Ansichten $j$ und $k$ dar. Er ergibt sich aus den drei obengenannten Gleichungen und ist gegeben durch:

$$\gamma_k = \frac{(\boldsymbol{t}_{jk} \wedge \boldsymbol{t}_{ik})^\top (\boldsymbol{t}_{jk} \wedge \boldsymbol{R}_{jk} \boldsymbol{t}_{ij})}{\|\boldsymbol{t}_{jk} \wedge \boldsymbol{t}_{ik}\|^2} \tag{4.22}$$

Wenn der Skalierungsfaktor bestimmt ist, wird die Kameraorientierung mit nicht linearen Verfahren optimiert.

### Optimierung

Die Kamerabewegung wurde mit Hilfe von linearen Verfahren bestimmt. Die Lösung kann jedoch auf Grund von Fehlern der Eingabedaten oder falschen Punktzuordnungen sehr ungenau werden und sollte mit nichtlinearen Optimierungsverfahren verfeinert werden.
Die Fehlerfunktion, über die die Variablen optimiert werden, minimiert den Abstand der Punkte zu ihren epipolaren Linien. Einen wesentlichen Arbeitsschritt stellt hierbei die Parametrisierung der Rotationen und der Translationen dar.
Die Rotation hängt von drei Parametern ab. Eine Parametrisierung über die drei Rotationswinkel ist möglich, beinhaltet aber Singularitäten für die Winkelwerte 0 und $\pi$. Eine Alternative stellt eine Parametrisierung mit Hilfe eines Quaternions $(\boldsymbol{l}, \Omega)$ dar, wobei durch $\boldsymbol{l}$ die Rotationsachse ($\|\boldsymbol{l}\| = 1$) und durch $\Omega$ der Rotationswinkel beschrieben wird.
Der Translationsvektor zwischen den zwei ersten Ansichten ist ein dreidimensionaler Vektor mit einer festen Länge, die auf den Wert eins gesetzt wird. Daraus folgt, dass $\boldsymbol{t}$ mit $3 - 1 = 2$ Parametern beschrieben werden kann. Für zwei Ansichten werden insgesamt fünf Parameter optimiert.
$\boldsymbol{m}_{i1}$ und $\boldsymbol{m}_{i2}$ seien zwei Punkte der ersten bzw. der zweiten Ansicht, und $\boldsymbol{l}_{m'_{i1}}$ und $\boldsymbol{l}_{m'_{i2}}$ die entsprechenden, zugehörigen epipolaren Linien. Die Fehlerfunktion $\boldsymbol{F}(\boldsymbol{t}, \boldsymbol{l}, \Omega)$ ist als Summe der Abstände $d$ der Punkte zu den epipolaren Linien definiert:

$$\boldsymbol{F}(\boldsymbol{t}, \boldsymbol{l}, \Omega) = \sum_{i=1}^{n} (d(\boldsymbol{m}_{i2}, \boldsymbol{l}_{m'_{i1}})^2 + d(\boldsymbol{l}_{m'_{i2}}, \boldsymbol{m}_{i1})^2) \tag{4.23}$$

Bei drei Ansichten ist die Anzahl der zu bestimmenden Parameter nicht $3 \times 5 = 15$ sondern $3 \times 2 + 2 \times 2 + 1 = 11$. Da die Parameter in gegenseitiger Abhängigkeit stellen, werden weniger

Parameter pro Bild benötigt. Auf Grund der geringeren Anzahl an Parameter verläuft die Bestimmung der relativen Bewegungen stabiler und genauere Ergebnisse können erreicht werden.
Bei falscher Zuordnung der Bildpunkte müssen robuste Methoden, wie die sogenannten M-Estimatoren-Methode, angewendet werden [69]. Sie minimieren den Einfluss sogenannter *Outliers*, indem sie bei der Optimierung dieser Punkte geringer gewichtet oder ignoriert werden.

### 4.4.4 Rekonstruktion

### 4.4.5 Rekonstruktionsverfahren

Nach der Bestimmung der relativen Bewegungen sind die Projektionsmatrizen der $n$-Bilder bekannt, wodurch eine Rekonstruktion der Szene ermöglicht wird.
$\boldsymbol{m}(m_x, m_y, 1)$ sei die Projektion eines 3D-Punktes $\boldsymbol{M}(X, Y, Z, 1)$. Es gilt:

$$\boldsymbol{m} \sim \boldsymbol{PM} \tag{4.24}$$

mit $\boldsymbol{P} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)^\top$, wobei die Vektoren $\boldsymbol{p}_i$ $(i = 1, 2, 3)$ die Zeilenvektoren der Matrix $\boldsymbol{P}$ darstellen.
Nach Eliminierung des Skalarfaktors wird die Gleichung 4.24 wie folgt geschrieben:

$$\boldsymbol{p}_1\boldsymbol{M} - m_x.\boldsymbol{p}_3\boldsymbol{M} + p_{14} - m_x p_{34} = 0 \tag{4.25}$$
$$\boldsymbol{p}_2\boldsymbol{M} - m_y.\boldsymbol{p}_3\boldsymbol{M} + p_{24} - m_y p_{34} = 0 \tag{4.26}$$

Für die $n$-Bilder resultiert aus diese Formulierung ein lineares Gleichungssystem :

$$\boldsymbol{AM} = \boldsymbol{b} \tag{4.27}$$

wobei $\boldsymbol{A}$ eine $2n \times 3$ Matrix beschreibt. Die Matrix $\boldsymbol{A}$ ist folgendermaßen definiert:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{p}_{10} - m_{x1}\boldsymbol{p}_{10} \\ \boldsymbol{p}_{11} - m_{y1}\boldsymbol{p}_{12} \\ \vdots \\ \boldsymbol{p}_{n0} - m_{xn}\boldsymbol{p}_{n2} \\ \boldsymbol{p}_{n1} - m_{yn}\boldsymbol{p}_{n2} \end{pmatrix} \tag{4.28}$$

und

$$\boldsymbol{b} = \begin{pmatrix} m_{x1}(p_{34})_1 - (p_{14})_1 \\ m_{y1}(p_{34})_1 - (p_{24})_1 \\ \vdots \\ m_{xn}(p_34)_n - (p_{14})_n \\ m_{yn}(p_34)_n - (p_{24})_n \end{pmatrix} \tag{4.29}$$

Das System $\boldsymbol{AM} = \boldsymbol{b}$ wird mit Hilfe des SVD-Verfahrens gelöst [69].

**Berechnung der Kovarianz**

Nach der Berechnung der Koordinaten der 3D-Punkte werden die jeweiligen Kovarianzmatrizen ermittelt und dadurch eine Schätzung der Genauigkeit der Rekonstruktion gegeben. Die Berechnung der Kovarianzmatrizen erfolgt auf Basis des SVD-Verfahrens, mit dem die Punkte, wie im Abschnitt 4.4.4 erläutert, rekonstruiert wurden.
Die Kovarianzmatrix $\boldsymbol{C}$ eines 3D-Punktes $\boldsymbol{M}$ ist folgendermaßen definiert:

$$\boldsymbol{C} = \sum_{i=1}^{3} \frac{1}{w_i^2} \boldsymbol{V}_i \boldsymbol{V}_i^\top$$

mit $\boldsymbol{V}$ als Matrix der SVD-Zerlegung von $\boldsymbol{A}$: $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}(w_i)\boldsymbol{V}^\top$.
Eine Darstellung der Genauigkeit der Rekonstruktion wird durch die Konfidenzellipsen ($\Delta\chi = 1$) ermöglicht, siehe [69]. Für jeden Punkt können die drei Achsen der Konfidenzellipse in das 3D-Modell eingeblendet werden und dadurch die Güte der Rekonstruktion intuitiv beurteilt.

### 4.4.6 Verdeckungsbehandlung

Für die Behandlung der Verdeckungen ist ein 3D-Modell der Szene notwendig. Im Allgemeinen muss das Modell muss nicht sehr komplex sein und kann auf einzelne Polygone in der direkten Umgebung des virtuellen Objektes beschränkt werden.
Dies ist nicht der Fall, wenn beispielsweise die Lichtverhältnisse zwischen realen und virtuellen Objekten simuliert werden müssen. Dann muss theoretisch der gesamte Raum auf Grund eventueller Reflektionen oder im Bild nicht sichbarer Lichtquellen modelliert werden. Eine zusätzliche Erweiterung des Szenemodells besteht darin, physikalisches Verhalten wie z.B. Schwerkraft, Kollision, u.ä., zu ermöglichen [39].

## 4.5 Calibration Propagation

### 4.5.1 Von einem kalibrierten zu einem unkalibrierten Bild

Der Term *Calibration Propagation* wurde in [26] eingeführt und bezeichnet die Konfiguration von zwei Bildern, wobei die intrinsischen Kameraparameter nur für ein Bild bekannt sind. Auf Basis von Punktkorrespondenzen werden die relativen Bewegungen und die intrinsischen Parameter der zweiten Kamera berechnet. Dieses Verfahren wurde *Calibration Propagation* genannt, da die Kalibrierungsinformationen von einem kalibrierten auf ein unkalibriertes Bild übertragen werden. Mit dem Verfahren *Calibration Propagation* ist es möglich, Zoomänderungen und Kamerabewegungen gleichzeitig zu erfassen.

### 4.5.2 Die Matrix $\boldsymbol{Q}$

Für zwei gegebene Bilder wird die fundamentale Matrix $\boldsymbol{F}$, wie im Abschnitt 3.3.3 erläutert, berechnet. Es wird vorausgesetzt, dass die Matrix der intrinsischen Parameter $\boldsymbol{A}_1$ des ersten Bildes bekannt sind.
Eine neue Matrix $\boldsymbol{Q}$ wird eingeführt und wie folgt definiert:

$$\boldsymbol{Q} = \boldsymbol{F}\boldsymbol{A}_1 \tag{4.30}$$

Auf Grund der Definition der fundamentalen Matrix, siehe Kapitel 3, kann $\boldsymbol{Q}$ folgendermaßen geschrieben werden:

$$\boldsymbol{Q} = \boldsymbol{FA} \sim \boldsymbol{A}_2^{-\top}[\boldsymbol{t}]_\wedge \boldsymbol{R} \tag{4.31}$$

$\boldsymbol{Q}$ definiert die epipolare Geometrie zwischen dem ersten, normalisierten (metrischen) Bild und dem zweiten, unkalibrierten Bild.

### 4.5.3 Intrinsische Parameter und Translationsvektoren

Um die intrinsischen Parameter der zweiten Kamera, d.h. die Matrix $\boldsymbol{A}_2$, zu bestimmen, wird zuerst der Produkt $\boldsymbol{QQ}^\top$ gebildet und dadurch die Rotationsmatrix $\boldsymbol{R}$ eliminiert.
Das Produkt $\boldsymbol{QQ}^\top$ ist wie folgt definiert:

$$\boldsymbol{QQ}^\top \sim \boldsymbol{A}_2^{-\top}[\boldsymbol{t}]_\wedge^2 \boldsymbol{A}_2^{-\mathbf{1}} \tag{4.32}$$

$\boldsymbol{QQ}^\top$ hängt nur von der Matrix $\boldsymbol{A}_2$ und dem Vektor $\boldsymbol{t}$ ab, d.h. die Matrix $\boldsymbol{QQ}^\top$ wird mit Hilfe von sechs Parametern definiert; vier für die Matrix der Kameraparameter $\boldsymbol{A}_2$ und zwei für den Translationsvektor $\boldsymbol{t}$, da die Länge des Translationsvektors arbiträr ist und $\|\boldsymbol{t}\| = 1$ gesetzt wird.
Die Matrix $\boldsymbol{QQ}^\top$ ist eine symmetrische Matrix. Dies bedeutet, dass maximal fünf unabhängige Gleichungen existieren und nicht alle Parameter $(\boldsymbol{A}_2, \boldsymbol{t})$ aus nur zwei Bildern berechnet werden können.
Im Fall einer Änderung durch Zoomen kann angenommen werden, dass sich nur ein intrinsischer Parameter, nämlich die Brennweite, verändert. In diesem Fall beschränken sich die zu extrahierenden Parameter auf die zwei Translationskomponenten und die Brennweite. Bei dieser Approximation wird der Aspekt-Ratio gleich 1 gesetzt und das optische Kamerazentrum auf den Bildmittelpunkt festgelegt.
Die Matrix $\boldsymbol{A}_2$ kann dann folgendermaßen geschrieben werden:

$$\boldsymbol{A}_2 = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.33}$$

$\boldsymbol{QQ}^\top$ ist eine Funktion des Translationsvektors $\boldsymbol{t} = (x, y, z)$ und der Brennweite $f$ und ist gegeben durch:

$$\boldsymbol{QQ}^\top \sim \begin{pmatrix} -\frac{y^2+z^2}{f^2} & \frac{xy}{f^2} & \frac{xz}{f} \\ \frac{xy}{f^2} & -\frac{x^2+z^2}{f^2} & \frac{zy}{f} \\ \frac{xz}{f} & \frac{zy}{f} & -(x^2+y^2) \end{pmatrix} \tag{4.34}$$

Aus Gleichung 4.34 werden die einzelnen Komponenten von $\boldsymbol{QQ}^\top = (q_{ij}), (i, j = 1, \ldots, 3)$ wie folgt beschrieben:

$$q_{11} = -\frac{y^2+z^2}{f^2} \tag{4.35}$$
$$q_{12} = \frac{xy}{f^2} \tag{4.36}$$
$$q_{13} = \frac{xz}{f} \tag{4.37}$$
$$q_{22} = -\frac{x^2+z^2}{f^2} \tag{4.38}$$
$$q_{23} = \frac{zy}{f} \tag{4.39}$$
$$q_{33} = -(x^2+y^2) \tag{4.40}$$

Die Translationskomponenten werden ausgeschlossen und die Gleichung 4.38 ausgewertet. Die Brennweite $f$ kann aus der Gleichung direkt extrahiert werden und ist, wie folgt, definiert:

$$f = |\frac{q_{23}q_{13}}{q_{12}}|\sqrt{\frac{q_{22}-q_{11}}{q_{11}q_{13}^2 - q_{22}q_{23}^2}} \tag{4.41}$$

Eine weitere Möglichkeit besteht darin, die letztgenannte Gleichung 4.40 ($q_{33} = -(x^2+y^2)$) zu verwenden, um $f$ zu bestimmen. Daraus ergibt sich eine Gleichung zweiter Ordnung:

$$f^4 q_{22}(1 + (\frac{q_{23}}{q_{13}})^2) - q_{33}f^2 - (\frac{q_{23}}{q_{12}})^2 q_{33} = 0 \tag{4.42}$$

Anschließend kann problemlos der Translationsvektor $\boldsymbol{t}$ aus Gleichung 4.32 abgeleitet werden.
Die Berechnung der Rotation $\boldsymbol{R}$ erfolgt, wie in Abschnitt 3.4.3 beschrieben, aus der essentielle Matrix $\boldsymbol{E}$, wobei $\boldsymbol{E} = \boldsymbol{A}_2{}^\top \boldsymbol{Q}$ gilt.

### 4.5.4 Nichtlineare Optimierung

Die Ergebnisse der Brennweite $f$ und des Translationsvektors $\boldsymbol{t}$ aus Gleichung 4.41 werden mit einem nichtlinearen Optimierungsverfahren verfeinert. Folgende Fehlerfunktion $\boldsymbol{F}(f, \boldsymbol{t})$ wird minimiert:

$$min_{f,\boldsymbol{t}} \|\boldsymbol{F}(f,\boldsymbol{t})\|^2 = min_{f,\boldsymbol{t}} \|\boldsymbol{Q}\boldsymbol{Q}^\top - \boldsymbol{A}_2^{-\top}[\boldsymbol{t}]_\times^2 \boldsymbol{A}_2^{-1}\|^2 \tag{4.43}$$

Anhand der in Abschnitt 4.5.7 beschriebenen Versuchen kann festgestellt werden, dass die Startwerte von $f$ un $\boldsymbol{t}$ nur wenig von der richtigen Lösung abweichen und deswegen nur wenige Iterationen zur Konvergenz der nichtlinearen Minimierung nötig sind.

### 4.5.5 Verwendung von 3D-Informationen

Um die intrinsischen Parameter der zweiten Kamera abzuleiten, können auch 3D- Informationen der Szene genutzt werden. Dazu wird zunächst eine projektive Rekonstruktion vorgenommen und anschließend die Transformation zwischen projektivem und euklidischem Raum mit Hilfe von 3D-Kenntnissen, wie z.B. Koordinaten von 3D-Punkten, Linien oder Ebenen, bestimmt.
Im folgenden wird zuerst der Zusammenhang zwischen der fundamentalen Matrix und den Projektionsmatrizen beider Bilder im projektiven Raum erstellt und anschließend erläutert, wie mit Hilfe von 3D-Daten die Projektionsmatrizen im euklidischen Raum bestimmt werden können.

**Projektiver Raum**

Im projektiven Raum werden die Projektionsmatrizen direkt aus der fundamentalen Matrix $\boldsymbol{F}$ berechnet. In Kapitel 3 wurde bewiesen, dass die Matrix $\boldsymbol{F}$ als eine Funktion des Epipoles $\boldsymbol{e'}$ des zweiten Bildes und einer Homographie $\boldsymbol{H}_\pi$ einer Ebene der Szene [60] wie folgt darstellbar ist:

$$\boldsymbol{F} = [\boldsymbol{e'}]_\wedge \boldsymbol{H}_\pi \tag{4.44}$$

Die Projektionsmatrizen beider Bilder können dann folgenderweise beschrieben werden:

$$\boldsymbol{P}_1 = (\boldsymbol{I}, \boldsymbol{0}) \tag{4.45}$$
$$\boldsymbol{P}_2 = (\boldsymbol{H}_\pi, \boldsymbol{e'}) \tag{4.46}$$

Aus den Matrizen $\boldsymbol{P}_1$ und $\boldsymbol{P}_2$ ergibt sich die fundamentale Matrix, die in Gleichung 4.44 definiert wurde.
Die Projektionsmatrizen müssen jedoch im euklidischen Raum ausgedrückt werden, um eine Erweiterung der Bilder mit virtuellen Objekten zu ermöglichen. Im nächsten Abschnitt wird gezeigt, wie mit Hilfe einer 3D-linearen Transformation (Kollineation) die gesuchten Matrizen bestimmt werden können.

### 4.5.6 Vom projektiven zum euklidischen Raum

Die Gleichung (4.44) ist auch für die Homographie der uneigentlichen Ebene (*plane at infinity*) $\boldsymbol{H}_\infty$ gültig. Daraus folgt, dass alle Homographien der Gleichung (4.44) durch eine Funktion von $\boldsymbol{H}_\infty$ mit einer beliebigen Matrix orthogonal zur Matrix $[\boldsymbol{e'}]_\wedge$ [60] darstellbar sind. Alle Matrizen $\boldsymbol{H}_\pi$ sind durch $\boldsymbol{H}_\infty$ in Kombination mit einem beliebigen Vektor $\boldsymbol{a}$ definiert:

$$\boldsymbol{H}_\pi \sim \boldsymbol{H}_\infty + \boldsymbol{e'}\boldsymbol{a}^\top \tag{4.47}$$

Darüber hinaus kann die Projektionsmatrix $\boldsymbol{P}_2$, da $\boldsymbol{H}_\infty = \boldsymbol{A}_2\boldsymbol{R}\boldsymbol{A}_1^{-1}$ gilt, folgendermaßen beschrieben werden:

$$\boldsymbol{P}_2 = (\boldsymbol{H}_\pi, c\boldsymbol{e}_2) = (\boldsymbol{A}_2\boldsymbol{R}\boldsymbol{A}_1^{-1} + \boldsymbol{e}_2\boldsymbol{a}^\top, c\boldsymbol{e}_2) \tag{4.48}$$

bzw.:

$$\boldsymbol{P}_2 = \boldsymbol{A}_2(\boldsymbol{R}, \boldsymbol{t}) \begin{pmatrix} \boldsymbol{A}_1^{-1} & \boldsymbol{0} \\ \boldsymbol{a} & c \end{pmatrix} \tag{4.49}$$

Daraus folgt:

$$\boldsymbol{H}_e = \begin{pmatrix} \boldsymbol{A}_1^{-1} & \boldsymbol{0} \\ \boldsymbol{a} & c \end{pmatrix} \tag{4.50}$$

Für einen Punkt im projektiven Raum $\boldsymbol{M}_p$, der in $\boldsymbol{m}$ auf die Bildebene projiziert wird, kann die folgende Gleichung aufgestellt werden:

$$\boldsymbol{m} = \boldsymbol{P}_2\boldsymbol{M}_p = \boldsymbol{A}_2(\boldsymbol{R}, \boldsymbol{t})\boldsymbol{H}_e\boldsymbol{M}_p \tag{4.51}$$

Im euklidischen Raum ist die Projektion $\boldsymbol{m}$ des Punktes $\boldsymbol{M}_e$ gegeben durch:

$$\boldsymbol{m} = \boldsymbol{A}_2(\boldsymbol{R}, \boldsymbol{t})\boldsymbol{M}_e \tag{4.52}$$

Aus einem Vergleich der beiden letzten Gleichungen wird die Transformation vom $\boldsymbol{M}_p$ in $\boldsymbol{M}_e$ abgeleitet:

$$\boldsymbol{M}_e \sim \boldsymbol{H}_e \boldsymbol{M}_p \tag{4.53}$$

Da die Matrix $\boldsymbol{A}_1$ von $\boldsymbol{H}_e$ bereits bekannt ist, beschränkt sich die Bestimmung von $\boldsymbol{H}_e$ auf die Bestimmung des Vektors $\boldsymbol{a}$ und des Skalars $c$.
Wenn 3D-Punkte bekannt sind, können sie mit Hilfe von $\boldsymbol{P}_1$ und $\boldsymbol{P}_2$ im projektiven Raum rekonstruiert und in Gleichung 5.7 verwendet werden. Die Anzahl der Parameter, die zu berechnen sind, entspricht vier, d.h. der Vektor $\boldsymbol{a}$ und der Skalar $c$. Jeder 3D-Punkt liefert jedoch drei Gleichungen, wobei der Skalierungsfaktor eliminiert werden muss. Dies bedeutet, dass mindestens zwei Punkte bzw. sechs Gleichungen benötigt werden.
Wenn die Homographie $\boldsymbol{H}_e$ bekannt ist, kann $\boldsymbol{P}_2 = \boldsymbol{A}_2(\boldsymbol{R}, \boldsymbol{t})$ leicht abgeleitet werden. Letztlich werden die intrinsischen Parameter aus der Projektionsmatrix mit Hilfe einer $QL$-Zerlegung extrahiert.
Diese Lösung bietet den Vorteil, dass die Berechnung von $\boldsymbol{H}_e$ direkt sowohl alle intrinsischen Parameter als auch die Rotation und die skalierte Translation liefert.

### 4.5.7 Evaluierung des Verfahrens *Calibration Propagation*

Um die Genauigkeit des Verfahrens *Calibration Propagation* zu evaluieren, wurden Untersuchungen mit synthetischen Bildern und Standardbildern durchgeführt. Die Ergebnisse werden in den beiden folgenden Paragraphen vorgestellt.

#### Synthetische Bilder

Im folgenden Abschnitt wird das Verfahren *Calibration Propagation* weitergehend untersucht und evaluiert.
Bei den Untersuchungen mit synthetischen Bildern ist das optische Kamerazentrum auf dem Bildmittelpunkt gesetzt und der Aspekt-Ratio auf eins festgelegt. Aus zehn bis fünfzehn interaktiv selektierten Bildpunkten wird die Matrix $\boldsymbol{F}$ berechnet und anschließend die Brennweite für die Bilder abgeleitet. Die beiden folgenden Abbildungen illustrieren diesen Vorgang anhand Bilder der Szene “Room” 4.20 und “Lab” 6.18.

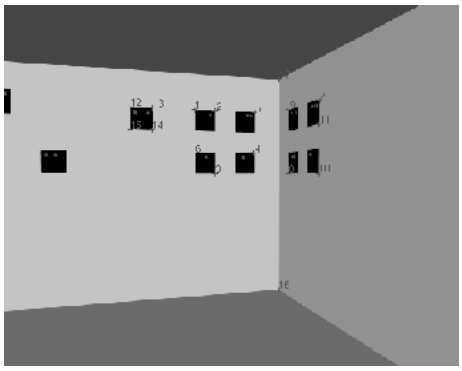
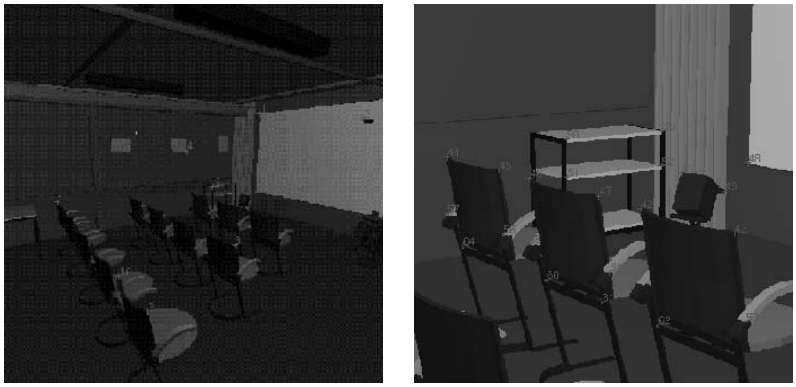


Abbildung 4.20: Synthetisches Testbild “Room” (R)



(a) Lab ($f = 420$) (b) Lab ($f = 1701$)

Abbildung 4.21: Evaluierung des *Calibration-Propagation*-Verfahrens anhand synthetischer Bilder "Lab"(L)

Um das Verfahren zu untersuchen, wurden zunächst drei Bilder der Szene "Room" generiert. Die Kamera besitzt dieselben intrinsischen Parameter und ist mit unterschiedlichen Orientierungen an verschiedenen Stellen im Raum positioniert.
Obwohl es sich um synthetische Bilder handelt, sind die Punktpositionen im Bild nicht-fehlerfrei, da sie, wie für reale Bilder, per Hand eingegeben wurden. Tabelle 4.3 zeigt die Ergebnisse der Schätzung der Brennweite für die Bilder R(1) und R(2). Die lineare Lösung liefert eine gute Approximation der richtigen Brennweite, die anschließend als Startwert für die nichtlineare Methode weiterverwendet wurde. Die Ergebnisse werden durch diese zweite Berechnung stark verbessert und es wird eine sehr gute Genauigkeit erreicht. Die relativen Fehler der Brennweite betragen nur 0.79% für R(1) und 2.41% für R(2) und sind klein genug, um eine gute Bildüberlagerung zu ermöglichen.

| Bild | Richtige Brennweite | Lineare Lösung | Nicht-lineare Optimierung | Relative Fehler (in %) |
|---|---|---|---|---|
| R(1) | 720 | 767.8 | 714.3 | 0.79 |
| R(2) | 720 | 653.0 | 737.4 | 2.41 |
| L(1) | 420.5 | 427.6 | 422.9 | 0.57 |
| L(2) | 420.5 | 424.2 | 421.4 | 0.21 |
| L(3) | 1119.6 | 1136.5 | 1125.2 | 0.50 |
| L(4) | 1701.4 | 1785.1 | 1673.6 | 1.63 |

Tabelle 4.3: Schätzung der Brennweite für die Szene "Lab" und "Room"

Bei der zweiten Versuchsreihe wurde das Verfahren mit vier Bildern der Szene "Lab" getestet. Das erste Bild wurde als Referenzbild gewählt. Das zweite Bild wurde mit einer Kamera mit derselben Brennweite und anderer Kameraposition und -orientierung erzeugt. Für das dritte und vierte Bild wurde sowohl die Kameraposition als auch die Brennweite verändert. Die Zoomfaktoren für das dritte und vierte Bilder sind relativ hoch und betragen ungefähr die Werte drei und vier, siehe Tabelle 4.3. Abbildung 6.18 zeigt das erste

L(1) und vierte L(4) Bild der Szene “Lab”.
Für die Szene “Lab” konnten mit der linearen Lösung sofort genaue Ergebnisse erzielt werden. Dieses Ergebnis kann hauptsächlich auf eine bessere Punktverteilung im Raum zurückgeführt werden. Daraus folgt eine bessere Schätzung der fundamentalen Matrix $\boldsymbol{F}$ und der Matrix $\boldsymbol{Q}$. Da die Berechnungen auf Basis dieser Matrizen erfolgen, wird auch eine genauere Ermittlung der Brennweite erzielt.
Insgesamt bleibt für alle Bilder “Room” und “Lab” der relative Fehler unter $2,5\%$. Diese guten Ergebnisse bestätigen die Gültigkeit des Verfahrens.

**Standardbilder**

Wie der vorangehende Abschnitt verdeutlicht hat, liefert das Verfahren *Calibration Propagation* gute Ergebnisse bei Verwendung synthetischer Bilder. Daher wurden weitergehend Untersuchungen mit Standardbildern durchgeführt.

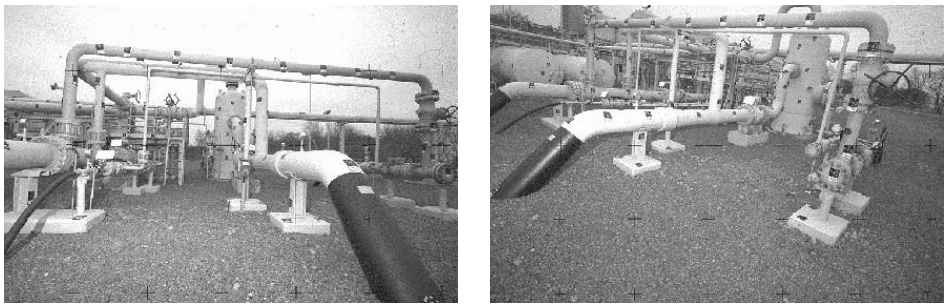


Abbildung 4.22: Zwei Testbilder einer industriellen Anlage

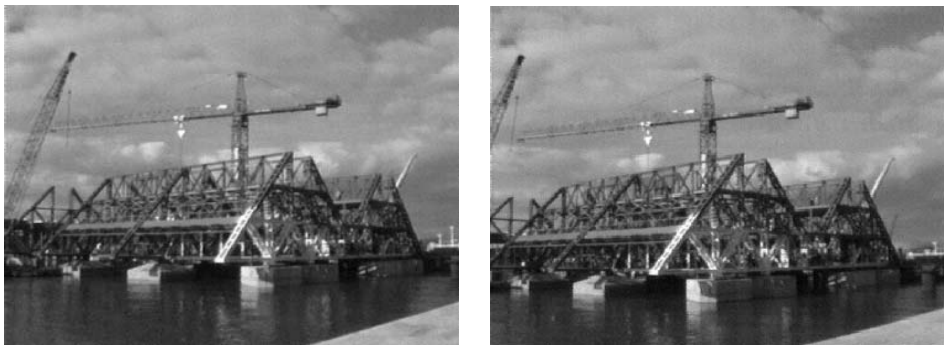


Abbildung 4.23: Testbilder “Expo”

Die Bilder in Abbildung 4.22 wurden mit einer professionellen Kamera aufgenommen. Die intrinsischen Parameter sind mit hoher Genauigkeit bekannt und dienen als Referenzparameter. Im Rahmen der Testserie werden sie für das zweite Bild mit Hilfe des ersten Bildes neu berechnet und verglichen.
Die Berechnung der Brennweite erfolgt in diesem Fall mit einem approximierten optischen Kamerazentrum, da dieses a priori unbekannt ist und aus diesem Grund im Bildmittelpunkt festgelegt wird.
Wie für die synthetischen Bilder werden die Eingabepunkte interaktiv in den Bilder eingegeben und daraus die fundamentale Matrix $\boldsymbol{F}$ und anschließend $\boldsymbol{Q}$ berechnet. Das Verfah-

ren liefert auch hier relativ genaue Ergebnisse. Trotz Approximation des Kamerazentrums beträgt der relative Fehler nur 2.5%.

| Referenz-parameter | Lineare Lösung | Nicht-lineare Optimierung | Relative Fehler (in %) |
|---|---|---|---|
| f = 400.36 | 388.7 | 390.3 | 2.5 |
| cx = 358.05 | 384 | 384 | |
| cy = 249.06 | 256 | 256 | |

Tabelle 4.4: Schätzung der Brennweite (Das optische Kamerazentrum ist nicht bekannt und im Bildmittelpunkt gelegt worden)

Um das Verfahren zu untersuchen, wurden die intrinsischen Parameter für die Bilder "Expo" mit Hilfe einer direkten Kalibrierung nach dem Tsai-Algorithmus [85] ermittelt. Die 3D-Koordinaten der Kalibrierungspunkte stammen aus einem CAD-Modell und wurden interaktiv im Bild eingegeben. Da die intrinsischen Parameter der Kamera ebenfalls berechnet wurden, stellen sie keine richtigen *Ground Truth*-Werte dar, sondern gute Annäherungen davon.

| Referenz-parameter | lineare Lösung | Nicht-lineare Optimierung | Relative Fehler (in %) |
|---|---|---|---|
| f = 979 | 860.54 | 1033.01 | 5.5 |
| cx = 283.7 | 384 | 360 | |
| cy = 287.5 | 256 | 288 | |
| sx = 0.874 | 1 | 1 | |
| sy = 1 | 1 | 1 | |

Tabelle 4.5: Berechnung von $f$ (Bilder "Expo")

Die Fehler, in Tabelle 4.5 zusammengestellt, sind, wie zu erwarten war, größer als bei allen vorherigen Untersuchungen. Im Vergleich beträgt der Fehler für die lineare Lösung ungefähr 12%, nach der Optimierung jedoch nur 5%.

### 4.5.8 Schlussfolgerung

Die *Calibration Propagation* Methode wurde anhand synthetischer sowie realer Bilder getestet. Die vorgeschlagene Lösung für die Berechnung der Brennweite liefert gute und stabile Ergebnisse mit einer Genauigkeit von 2 bis 5%. Dabei sind verschiedene Fehlerquelle zu unterscheiden.
1. Die Kalibrierung des ersten Bildes ist nicht exakt und fehlerfrei.
2. Aus der manuellen Punkteingabe in den Bildern resultiert eine mehr oder weniger große Ungenauigkeit der Lokalisierung.
3. Die Berechnung der fundamentalen Matrix $\boldsymbol{F}$ hängt von der Verteilung der Punkte in Raum und im Bild ab.
4. Letztendlich wird für das zweite Bild das optische Kamerabildzentrum und der Aspektratio approximiert.

## 4.6 Augmented Video

### 4.6.1 Vorgehensweise

Für die Bearbeitung von Videosequenzen müssen auf Grund der hohen Anzahl der Bilder Automatisierungsmechanismen eingeführt werden.
Das entwickelte Verfahren besteht aus den folgenden Schritten:

1. Zwei Referenzbilder werden aus der Videoabfolge selektiert. Üblicherweise wird das erste und letztes Bild gewählt.
2. Die Bildpunkte werden interaktiv in die Bilder eingegeben. Aus diesen zwei Bildern werden mit den im vorherigen Abschnitt beschriebenen Methoden die relative Kamerabewegung und die 3D-Punktkoordinaten abgeleitet.
3. Ein einfaches 3D-Modell der Szene wird erzeugt und das virtuelle Objekt in der Szene platziert.
4. Die ausgewählten Punkte werden iterativ über die ganze Sequenz verfolgt, wobei die Kameraposition und -orientierung für jedes Bild auf Basis der 3D-Koordinaten der Bildpunkte berechnet wird.
5. Eine Optimierung zur Verfeinerung der 3D-Koordinaten der Referenzpunkte und der Kameraposition und -orientierung erfolgt über die ganze Bildsequenz.
6. Das virtuelle Objekt wird in jedem Bild eingeblendet und das Ergebnis gespeichert.

### 4.6.2 2D-Punktverfolgung

Ein Mechanismus zur automatischen Punktverfolgung ist für die Bearbeitung von Videosequenzen auf Grund der hohen Anzahl der Bilder notwendig. Die Verfolgung der Bildpunkte wird hier mit einem Korrelationsoperator realisiert. Dieser Operator hat den Vorteil jeden beliebigen Punkt oder Bildausschnitt von einem Bild zum nächsten lokalisieren zu können. Eine allgemeine Definition der Korrelation zweier Vektoren ist im folgenden Abschnitt gegeben.

#### Korrelationsoperator

Die Korrelation $\rho_{xy}$ zwischen zwei Vektoren $\boldsymbol{x}(x_1, \ldots, x_n)$ und $\boldsymbol{y}(y_1, \ldots, y_n)$ ist folgendermassen definiert [69]:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

wobei: $\sigma_x, \sigma_y$ die Standardabweichungen der Vektoren $\boldsymbol{x}$ und $\boldsymbol{y}$ darstellt und $\sigma_{xy}$ dem Kovarianzkoeffizient entspricht.
Die Korrelation zwischen dem Punkt $\boldsymbol{m}(u_0, v_0)$ in Bild 1 und dem Punkt $\boldsymbol{m'}(u_0 + \alpha, v_0)$ im Bild 2 über ein Fenster mit Dimensionen $(2N + 1, 2P + 1)$ ist definiert durch:

$$C(\alpha) = \frac{1}{K} \sum_{u_1=-N}^{+N} \sum_{u_1=-P}^{+P} (I_1(u_0 + u_1, v_0 + v_1) - I_1(\bar{u_0}, v_0))(I_2(u_0 + u_1 + \alpha, v_0 + v_1) - I_2(u_0 + \alpha, v_0)) \tag{4.54}$$

wobei $K = (2N+1)(2P+1)\sigma_1(u_0, v_0)\sigma_2(u_0+\alpha, v_0)$ gilt.

### Korrelation mit Sub-Pixel-Genauigkeit

Die Präzision, mit der die Punkte im Bild lokalisiert werden, ist für das Tracking und die Qualität der Video-Erweiterung entscheidend. Rauschen oder eine ungenaue Lokalisierung haben starke Auswirkungen auf die Bestimmung der Kameraposition und -orientierung. Aus diesem Grund wird eine Methode zur Punktverfolgung per Korrelation mit Sub-Pixel-Genauigkeit eingeführt. Verschiedene Verfahren sind möglich und werden beispielsweise in [54, 72] verglichen. In dieser Arbeit wird ein Verfahren, das auf die Interpolation der Pixelwerte mit einer bilineare Methode basiert, auf Grund seiner einfachen Implementierung ausgewählt.
Die Punktverfolgung besteht letzendlich aus zwei Schritten. Zuerst werden die Punkte mit dem Korrelationsoperator grob lokalisiert und anschließend mit der aufwendigeren, aber genaueren Sub-Pixel-Methode die Lokalisierung präzisiert. In dem in Abschnitt 4.6.5 beschriebenen Experiment werden die Bildausschnitte mit einem Fakor 4 interpoliert und hochskaliert.

### Punktverfolgung mit dem Korrelationsoperator

Die folgende Darstellung illustriert die einzelnen Schritte bei einer Punktverfolgung mit Hilfe des Korrelationsoperators.

Abbildung 4.24: Automatische Punktverfolgung

Im ersten Bild wird eine Schablone mit Dimensionen $(2N+1, 2P+1)$ um einen gegebenen Punkt definiert. Mit Hilfe des Korrelationsoperators wird diese Schablone im nächsten Bild gesucht. Dafür wird in dem zweiten Bild ein Suchbereich $S$ festgelegt und für jede Pixelposition der Korrelationswert mit der Schablone des ersten Bildes berechnet. Die neue Position der Schablone wird durch die Position des Maximums aller Korrelationswerte festgelegt.

### 4.6.3 Kamerabewegung

Die 3D-Koordinaten der verfolgten Punkte und die Kameraposition und -orientierung des ersten Bildes sind durch die interaktive Initialisierung, siehe Abschnitt 4.6.1, bekannt. Darüber hinaus ist die Kamerabewegung zwischen den beiden folgenden Bildern der Videosequenz sehr klein. Deswegen kann die Position $\boldsymbol{t}$ und Orientierung $\boldsymbol{R}$ der Kamera zwischen den Bildern iterativ berechnet werden, in dem als Startwerte die Werte des vorherigen Bildes benutzt werden.
Die Fehlerfunktion $\boldsymbol{F}(\boldsymbol{R},\boldsymbol{t})$, die hierfür angewendet wird, besteht aus dem Abstand zwischen den in das Bild re-projizierten 3D-Punkt und den in der Bildebene verfolgten Ausgangspunkt. $\boldsymbol{M}_i$ sei ein 3D-Punkt und $\boldsymbol{m}_i(u_i, v_i, 1)$ der zugehörige im Bild verfolgte 2D-Punkt $(i = 1, \ldots, n)$. Dann ist die Minimierung der Fehlerfunktion folgendermaßen definiert:

$$min_{\boldsymbol{R},\boldsymbol{t}}\|\boldsymbol{F}(\boldsymbol{R},\boldsymbol{t})\|^2 = min_{\boldsymbol{R},\boldsymbol{t}}\sum_{i=1}^{n}(u_i - \frac{\boldsymbol{p}_0^\top \boldsymbol{M}}{\boldsymbol{p}_2^\top \boldsymbol{M}})^2 + (v_i - \frac{\boldsymbol{p}_1^\top \boldsymbol{M}}{\boldsymbol{p}_2^\top \boldsymbol{M}})^2 \tag{4.55}$$

wobei die Vektoren $\boldsymbol{p}_0, \boldsymbol{p}_1$ und $\boldsymbol{p}_2$ die Zeilenvektoren der Projektionsmatrix $\boldsymbol{P} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2)^\top$ darstellen.

### 4.6.4 Reprojektion

Nach der Bestimmung der Kameraposition und -orientierung werden die 3D-Punkte in das nächste Bild projiziert. Anschließend wird die genaue 2D-Position der Punkte mit Hilfe des Korrelationsoperators bestimmt.
Diese Reprojektionstechnik bietet den Vorteil, dass, auch wenn ein Referenzpunkt über mehrere Bilder nicht verfolgt werden konnte, die initiale Suchposition und der Suchbereich immer richtig definiert sind, da sie aus der 3D-Kameraposition und -orientierung berechnet werden. Wenn im Vergleich die Punktverfolgung nur in 2D erfolgt, befindet sich der gesuchte Punkt nach Ausfall der Verfolgung über mehrere Bilder außerhalb des letzten gültigen Suchbereichs und wird keine oder falsche Tracking-Daten, d.h. Punktkoordinaten, liefern.

### 4.6.5 Bundle-Adjustment

Nachdem die Kameraposition und -orientierung für die ganze Videosequenz bestimmt wurde, wird eine Verfeinerung mit Hilfe eines Bundle-Adjustments vorgenommen. Die 3D-Koordinaten der Referenzpunkte werden mit Hilfe aller Bilder bestimmt und anschließend die Kameraposition und -orientierung nach der Gleichung 4.55 verfeinert.
Dabei ist jedoch zu berücksichtigen, dass für eine gegebene Videosequenz nicht immer alle Punkte mit dem Korrelationsoperator korrekt verfolgt werden. Auf Grund beispielsweise starker Verschiebungen zwischen einem Bild zum Nächsten oder auf Grund starkes Bildrauschens werden einige Punkte in den Bilder falsch lokalisiert. Diese Punkte stellen Ausreißer (outliers) dar und führen zu starken Fehler in der Berechnung der Kameraposition und -orientierung. Daraus resultiert, dass die eingeblendeten virtuellen Objekten in den Bildern falsch positioniert werden und über die ganze Videosequenz starke Sprünge oder "jitter" auftreten.
Auf Grund der fehlerhaften Daten wurde hierfür eine robuste Methode, Tuckey-Estimator genannt, angewendet. Diese Methode wurde ausgewählt, weil sie Ausreißer erkennt und de-

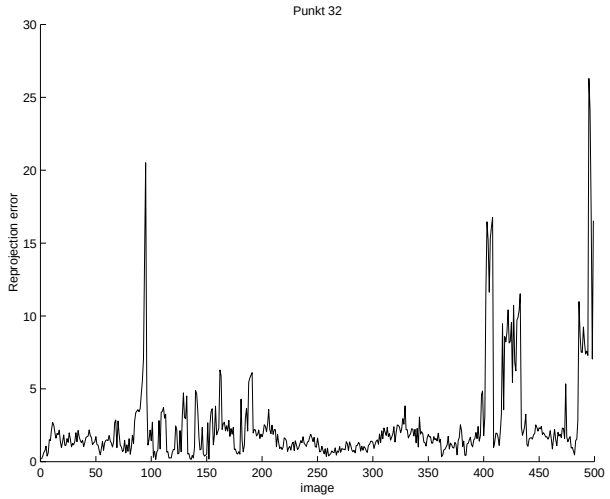



Abbildung 4.25: Reprojektionsfehler eines verfolgten Punktes (Bildsequenz von 500 Bildern)

ren Einfluß vermindert oder ausschaltet [62]. Abbildung 4.25 stellt die Reprojektionsfehler für einen Punkt der Videosequenz, siehe Abbildung 4.26 dar.
it Ausnahme einiger, weniger Bilder liegen generell nur kleine Fehler im Bereich von 2 bis 3 Pixel Abweichung vor. Für die Bilder mit Fehler von bis 20 Pixel (z.B. 100 und 410) würden ohne die robuste Tuckey-Estimator-Methode eine völlig falsche Kameraposition und -orientierung berechnet werden. Abbildung 4.26 zeigt das Endergebnis und die präzise Einblendung des virtuellen Modells über die ganze Bildsequenz.

## 4.7 Zusammenfassung

In diesem Kapitel wurden Lösungen zu Bild- und Videobearbeitung systematisch analysiert und vorgestellt. Die beschriebenen Methoden ermöglichen, alle notwendigen Informationen aus den Bildern zu extrahieren, wodurch auf 3D-Szene-Kentnisse verzichtet werden kann. Die theoretischen Grundlagen eines neuen Verfahrens, *Calibration Propagation* genannt, wurden vorgestellt und erörtert. Die praktische Umsetzung des Verfahrens wurde durch präzise Resultate belegt. Mit diesem Verfahren ist es möglich, sowohl Brennweite als auch Aufnahmeort der Kamera gleichzeitig zu verändern. Darüber hinaus wurde eine automatische Punktverfolgungsmethode für die Videobearbeitung präsentiert und implementiert. In Kombination mit robusten statistischen Minimierungsverfahren (M-Estimator) konnten lange Bildfolgen trotz fehlerhafter Daten mit virtuellen Objekten erweitert werden.

(0) (100) (200)

(300) (400) (500)

Abbildung 4.26: Bearbeitung einer Videosequenz (500 Bilder)

# Kapitel 5

# Markerbasiertes optisches Tracking

Einleitend wird in diesem Kapitel eines der ersten optischen Trackingsysteme für Augmented Reality vorgestellt. Die Algorithmen basieren auf einem "Punkt-zur-Linie"-Verfahren, das den Vorteil aufweist, wenig Rechner-Leistung zu benötigen und gleichzeitig eine hohe Genauigkeit zu liefern. Im zweiten Hauptabschnitt dieses Kapitels wird ein flexibles Tracking-System (VBT) beschrieben, das viele Funktionalitäten zur Bildverarbeitung und Mustererkennung anbietet und damit eine leichte Entwicklung und Integration neuer Verfahren ermöglicht. Zwei Lösungsansätze VBT-I und VBT-II, die mit schwarz-weißen und farbigen Marker arbeiten, werden anschließend präsentiert und evaluiert.

## 5.1 Das CVV-System

### 5.1.1 Grundlegende Konzepte

Das Echtzeit-Tracking stellt ein bisher ungelöstes Problem von Augmented-Reality dar. Die aktuell verfügbaren, kommerziellen Geräte sind auf die Anwendungsanforderungen der virtuellen Realität (VR) ausgerichtet und liefern nicht die für AR benötigte Genauigkeit. Darüber hinaus sind sie meist nicht tragbar und somit für den mobilen Einsatz ungeeignet. An die Entwicklung eines AR-Trackingsystems werden die vier folgenden Hauptanforderungen gestellt:

- einfache Handhabung,
- Echtzeit-Fähigkeit,
- hohe Genauigkeit,
- Mobilität und Tragbarkeit sowie
- Benutzung von Standard-Hardware-Komponenten.

Ein optisches und markerbasiertes System, das auf dem Inside-Out-Prinzip beruht, stellt in diesem Kontext die erfolgsversprechendeste Lösung dar. Das CVV-System (Computer Vision Viewer), das auf das Inside-Out-Prinzip aufbaut, wird in den folgenden Abschnitt

beschrieben. Dieses System stellt einen guten Kompromiss zwischen niedriger Rechnerkapazität, Echtzeit-Performance und Genauigkeit dar und ermöglicht die Realisierung und Evaluierung von innovativen AR-Anwendungen [23, 30].

### Das Inside-Out-Prinzip

Für das optische Tracking wurde der Inside-Out-Ansatz ausgewählt. Bei diesem Verfahren werden die Marker in der Szene und die Kamera am HMD des Benutzers angebracht. Wie im Kapitel 2 erläutert, können mit diesem Ansatz bei gleicher Tracking-Genauigkeit Kopfbewegungen wesentlich präziser als mit dem Outside-In-Verfahren erfasst werden.
Die Kamera soll, wie anhand des Bildes 5.1 gezeigt, möglichst nah am Display oder Auge montiert werden, um die natürliche Benutzersicht mit minimalem Offset erfassen zu können. Bei Mono-Video-Modus wurde experimentell festgestellt, dass sich die Kamera am besten in Blickmitte befinden soll.

Abbildung 5.1: HMD und Minikamera als Trackinggerät

Besonders vorzuheben ist, dass das Kamerabild gleichzeitig für das Tracking und die Bildüberlagerung benutzt werden kann. Somit wird nur ein Gerät auf dem HMD benötigt, so dass auch das Gewicht des HMD-Setups reduziert wird.

### Die Marker

#### Aktive versus passive Marker

Um die Bildverarbeitung zu vereinfachen und die Zuverlässigkeit des Trackers zu erhöhen, arbeiten viele optischen Mess- oder Tracking-Systeme mit aktiven Markern, wie beispielsweise LEDs (siehe Optotrack, Northern Digital Inc.) oder retroreflektiven Kugeln in Kombination mit Infrarot-Blitzen (QualiSys Inc., Vicon Inc., ART GmbH). Solche Marker können leicht vom Hintergrund unterschieden werden und liefern ein gut definiertes Bildsignal, das präzise zu erfassen ist. Dennoch sind diese Systeme für AR und besonders für mobile Anwendungen nicht geeignet, da sie Stationen im Raum voraussetzen. Zudem wird für die Erfassung der aktiven Marker spezielle und aufwändige Hardware benötigt.
Im Gegensatz dazu können passive Marker mit einer Standard-Kamera, die oft schon Bestandsteil eines AR-Systems ist, erfasst werden. Die Auswertung der Standard-Videobilder besitzt dabei vor allem den Vorteil, dass die Kontrolle der Lokalisierung direkt über das Bild, das überlagert wird, erfolgt und damit einerseits die Qualität der Überlagerung überprüft werden kann und andererseits keine weiteren Kalibrierungstransformationen zwischen Kamera und Tracker berechnet werden müssen.

**Entworfene Quadrat-Marker**
Die für das CVV-System entwickelten Marker bestehen aus einfachen schwarzen Quadraten auf weißem Grund. Ein Binärcode aus roten Punkten, siehe Abbildung 5.3, ermöglicht die eindeutige Identifizierung der Marker. Darüber hinaus liefert die quadratische Form vier Eckpunkte, die präzis zu erfassen sind und eine vollständige Bestimmung der Kameraposition und -orientierung ermöglichen.

### 5.1.2 Trackingablauf

Um hohe Echtzeit-Performance erreichen zu können, wurde der Trackingablauf in zwei Phasen, die in Abbildung 5.2 veranschaulicht sind, untergeteilt.

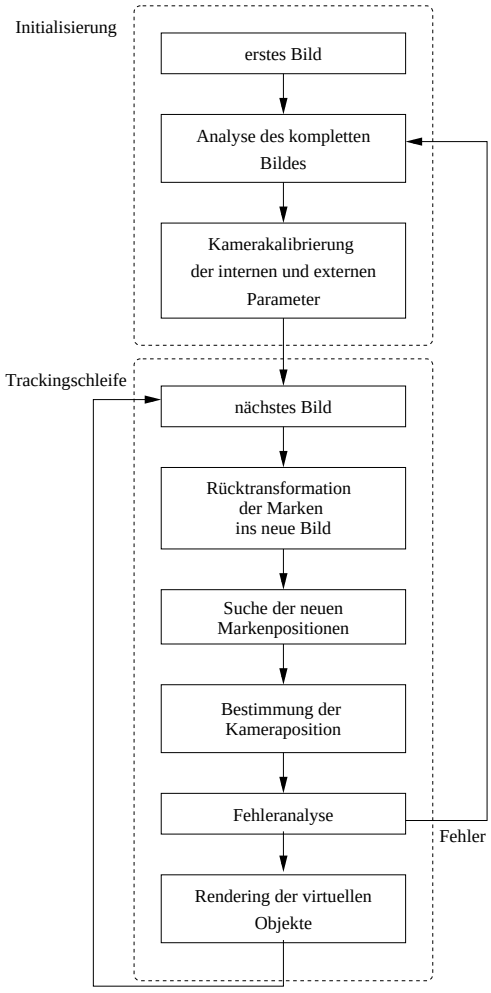



Abbildung 5.2: Blockdiagramm des CVV-Trackers

Zuerst erfolgt die Systeminitialisierung, in der das ganze Bild bearbeitet und die erste absolute Position der Kamera berechnet wird. Anschließend findet in der Trackingschleife eine lokale Bildanalyse und eine iterative Korrektur der Kameraposition statt.

### 5.1.3 Initialisierung

#### Markerdetektion

In der Initialisierungsphase wird zur Erkennung der Marker das komplette Bild analysiert. Jede $n$-te Zeile wird nach "hell-dunkel"- und "dunkel-hell"-Gradienten durchsucht, um mögliche Quadratkanten zu bestimmen. Ausgehend vom Zentrum der möglichen Quadratkanten wird senkrecht dazu nach einem dritten Punkt gesucht, wobei gleichzeitig die Homogenität und die Grauwerte der Pixel überprüft werden.

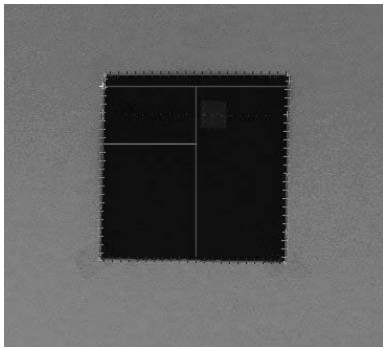


Abbildung 5.3: Markerdetektion

Treffen die Auswahlkriterien zu, wird anschließend die Kante der gefundenen schwarzen Region verfolgt und die Konturgradienten in vier Klassen, die der Kantenrichtungen eines Markers entsprechen, mit einem ISODATA-Algorithmus klassifiziert. Falls die vier Klassen eindeutig definiert sind, wird der Schnittpunkt der interpolierten Kantenlinien bestimmt. Sie stellen die Eckpunkte des Markers dar. In Abbildung 5.3 sind die Detektionsschritte, Suche nach starken Gradienten und Konturerfassung, veranschaulicht.

#### Identifikation

Bei der Identifikation werden die Markerkandidaten zunächst auf Größe und Form überprüft. Wenn dieser Test erfolgreich abgeschlossen ist, wird, um den Bezug zu den Punkten im Modell herzustellen, der Binärcode (ID) der einzelnen Marker anhand seiner roten Punkte bestimmt. Im abschließenden Identifikationsschritt wird ein Korrelationsoperator auf die gelesenen Pixelwerte und möglichen Identifikationsnummern angewendet. Der Marker ist identifiziert, wenn nur ein Korrelationskoeffizient existiert, der einen festen Schwellwert, 90% des maximalen Skors, überschreitet.
Anwendungen dieser Methode weisen eine hohe Robustheit auf und eignen sich auch bei der Verwendung Kameras niedriger Qualität, wie IndyCams, Webcams.

#### Kalibrierung

Um die Kamera zu kalibrieren, werden die Kameraparameter anhand der Eckpunkte der Marker mit Hilfe der Standard-Algorithmen von Tsai oder Weng [85, 90] ermittelt. Mit diesem Schritt ist die Initialisierung abgeschlossen und die Trackingschleife wird gestartet.

### 5.1.4 Trackingschleife

#### Rückprojektion und Prädiktion

Um den Marker-Suchbereich im Bild zu positionieren, wird das 3D-Modell der Marker entsprechend der vorherigen Kameraorientierung in das neue Bild zurückprojiziert. Anschließend erfolgt eine 2D-lineare Extrapolation der Position der Marker-Eckpunkte. Aufwendigere Prädiktionen, beispielsweise die Prädiktion in 3D mit einem Kalmanfilter [53], erweisen sich aufgrund der meist abrupten und unvorhersehbaren Kopfbewegungen des Benutzers nicht unbedingt als vorteilhaft. Der zusätzliche Rechneraufwand verlangsamt das gesamte System und ist insbesondere bei iterativen Verfahren unerwünscht, da die Abstände zwischen den Markern im Bild größer werden. Eine Diskussion über diese Problematik kann in [40] gefunden werden.

### 5.1.5 Markerdetektion

Um den Rechenaufwand zu begrenzen, können bei der Markerdetektion Bildoperationen minimiert werden, indem nur ein begrenzter Suchraum analysiert wird.

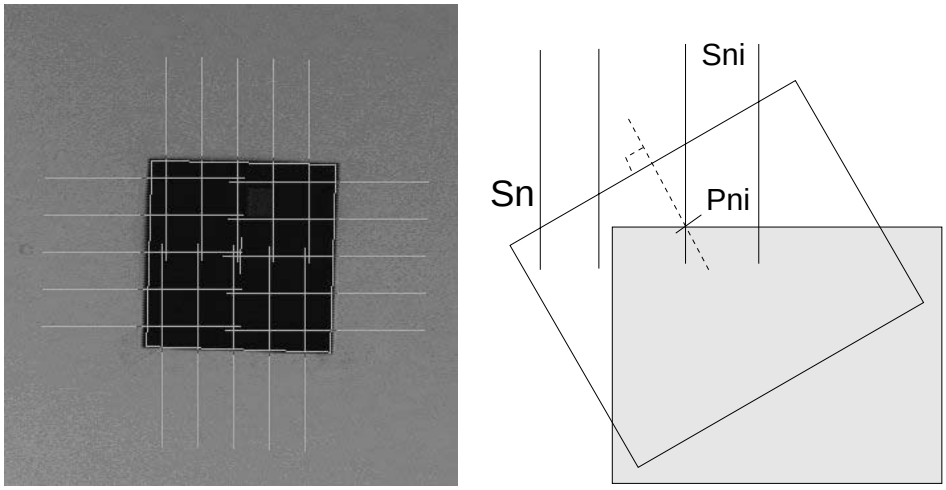



Abbildung 5.4: Marker und linearer Punkt-Suchbereich

Dafür werden die Segmente $\boldsymbol{S}_{ni}$, siehe Abbildung 5.4, senkrecht zu den Markerkanten definiert. Entlang dieser Segmente werden die Bildgradienten berechnet und das jeweils zum Segment gehörende Maximum gesucht. Die gefundenen Maxima stellen die Punkte $\boldsymbol{P}_i$ in Abbildung 5.4 dar. Die Präzision ihrer Lokalisierung wird mit Hilfe einer Interpolation zweiter Ordnung erhöht.
Zur Bestimmung der Kameraorientierung werden im folgenden Ansatz nur die Merkmale $\boldsymbol{P}_i$ verwendet, die entlang der Segmenten $\boldsymbol{S}_i$ lokalisiert wurden.

### 5.1.6 Bestimmung der Kamerabewegung: Das "Punkt-Linien-Verfahren"

Bei dem "Punkt-Linien-Verfahren" wird die Kameraorientierung in der Trackingschleife nicht neu berechnet, sondern iterativ von einem Bild zum Nächsten aktualisiert. Die Rotation und Translation der Kamera sind richtig geschätzt worden, wenn die rückprojizierten

Markerkanten $\boldsymbol{S}_n$ des 3D-Modells genau die detektierten Punkte $\boldsymbol{P}_{ni}$ überlagern. Das bedeutet, dass die Abstände $d_i$ von den Punkten $\boldsymbol{P}_{ni}$ zu den Markerkanten gleich null sein muss.
$\boldsymbol{U}_n$ sei der normale Vektor von $\boldsymbol{S}_n$ und $\boldsymbol{M}$ ein Punkt auf $\boldsymbol{S}_n$. Der Abstand $d_i$ ist dann wie folgt definiert:

$$d_i = \boldsymbol{U}_n.(\boldsymbol{P}_{ni} - \boldsymbol{M}) \tag{5.1}$$

Die Kamerarotation $\boldsymbol{R}$ und Translation $\boldsymbol{t}$ werden durch die Minimierung aller Abstände $d_i$ bestimmt. Mit diesem Ansatz wird die folgende Fehlerfunktion $F$ definiert:

$$F = \sum_{n=1}^{N} \sum_{i=1}^{S} \|\boldsymbol{U}_n.(\boldsymbol{P}_{ni} - \boldsymbol{M})\|^2 \tag{5.2}$$

wobei $N$ die Anzahl der Markerkanten im Bild und $S$ die Anzahl der Merkmale pro Kante bezeichnet.
Die Funktion $F$ wird mit Hilfe eines nichtlinearen Optimierungsverfahrens minimiert, das mit Hilfe des Levenberg-Marquardt-Verfahrens implementiert ist [69]. Um die Anfälligkeit des Verfahrens bezüglich falscher Daten zu reduziert, wurde der sogenannte "L1L2"-M-Estimator angewendet [62].

**Fehleranalyse**
Nach der Optimierung der Kameraparameter wird die Genauigkeit der Ergebnisse untersucht. Dafür werden die Fehler im Bild analysiert, indem der mittlere Abstand $(\boldsymbol{P}_{ni}, \boldsymbol{S}_n)$ und die entsprechende Standardabweichung gebildet werden. Überschreiten diese Werte vorgegebene Schwellwerten, konnte die Kameraorientierung nicht genau genug bestimmt werden. Die Trackingschleife wird dann verlassen und eine neue Initialisierung muss durchgeführt werden. Ansonsten wird das Verfahren mit dem nächsten Bild fortgesetzt.

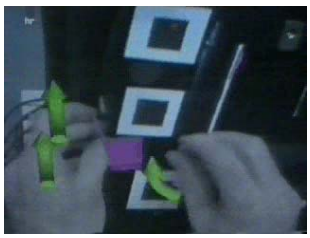
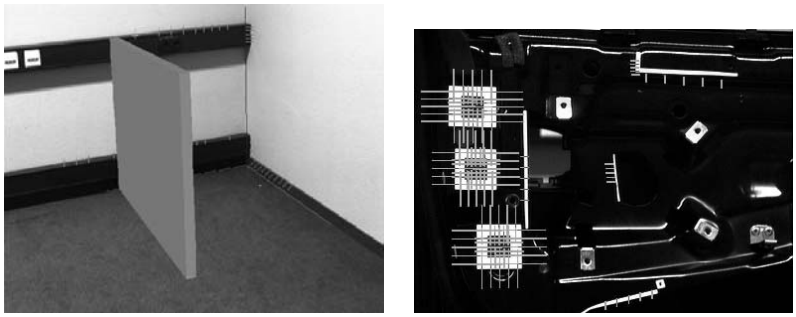


Abbildung 5.5: Robuste Markerdetektion

**Vorteil des Verfahrens**
Das vorgestellte Punkt-Linien-Verfahren weist den besonderen Vorteil auf, daß die Kameraorientierung direkt aus den Kantenpunkten $P_{ni}$ berechnet werden kann. Es handelt sich um eine robuste Minimierung über viele Merkmale, bei der falsche Daten einen minimalen Einfluss auf die Berechnung ausüben. Ein Beispiel wird in Abbildung 5.5 gegeben, bei dem die Marker zum Teil durch die Hände des Bnutzer verdeckt werden, aber dennoch eine genaue Überlagerung erreicht werden konnte.

### 5.1.7 Erweiterungen: Natürliche Szenemerkmale

Da die Kameraorientierung bei dem Punkt-Linien-Verfahren nur aus markanten Kanten gewonnen wird, kann diese Tracking-Methode sehr leicht durch natürliche Merkmale erweitert werden.



(a) (b)

Abbildung 5.6: Tracking natürlicher Merkmale (a) Ohne Marker (b) in Kombination mit Markern

In Abbildung 5.6(a) wird beispielsweise eine virtuelle Wand in einen Szenenbereich ohne Marker eingeblendet. Für viele praktische Anwendungen ist das markerlose Tracking leider dennoch nicht ausreichend stabil und besitzt noch zu viele Nachteile. So dürfen die Kamerabewegungen beim markerlosen Tracking nur sehr moderat sein, da der Suchbereich klein ist. Der Trackingsprozess ist rein iterativ, d.h. Marker werden für die Ermittlung der ersten Position- und Orientierungswerte benötigt. Dazu muss gewährleistet sein, dass alle Freiheitsgrade mit den Kanten im Bild der Kamera erfassen werden können und eine eindeutige Bestimmung von $\boldsymbol{R}$ und $\boldsymbol{t}$ ermöglichen.
Die Detektion und Verfolgung von natürlichen Kanten bietet jedoch in Kombination mit Markern eine sinnvolle Tracking-Unterstützung. Die Genauigkeit des Trackings wird deutlich erhöht, und eine korrekte Überlagerung kann auch in Bildbereichen ohne Marker erreicht werden. Darüber hinaus können kurze Verdeckungen aller Marker mit dem Tracking der natürlichen Kanten überbrückt werden. Bei beispielsweise dem Schlosseinbau einer Autotür, siehe Szenario im Kapitel 2 werden die Marker vom Monteur leicht verdeckt. Deswegen wurden zusätzlich als natürliche Kanten weiße Linien, wie in Abbildung 5.6(b) veranschaulicht, als zusätzliche Marker einbezogen.

### 5.1.8 Ergebnisse

Das Tracking-System CVV funktioniert in Echtzeit mit 20 bis 25 Hz auf einem Rechner mit niedriger Leistung (SGI O2, 180 MHz CPU). Es stellt eine Prototypisierung eines optischen Trackers für AR dar, mit dem es möglich ist, erste AR-Anwendungen zu realisieren und die Machbarkeit der Technologie zu evaluieren. Der Inside-Out-Ansatz mit einer am HMD montierten Standard-Kamera konnte als geeigneter Tracking-Ansatz für AR bestätigt werden. Ein großer Vorteil des CVV-Systems ist seine Robustheit gegenüber Verdeckungen. Diese Robustheit wird durch das Punkt-Linien-Verfahren, das im Abschnitt 5.1.6 beschrieben wird, erzielt.
Problematisch erweisen sich die hohen Geschwindigkeitsanforderungen an das System, aus deren Grund Marker nur lokal neu gesucht werden. Der Suchbereich entspricht hierbei der

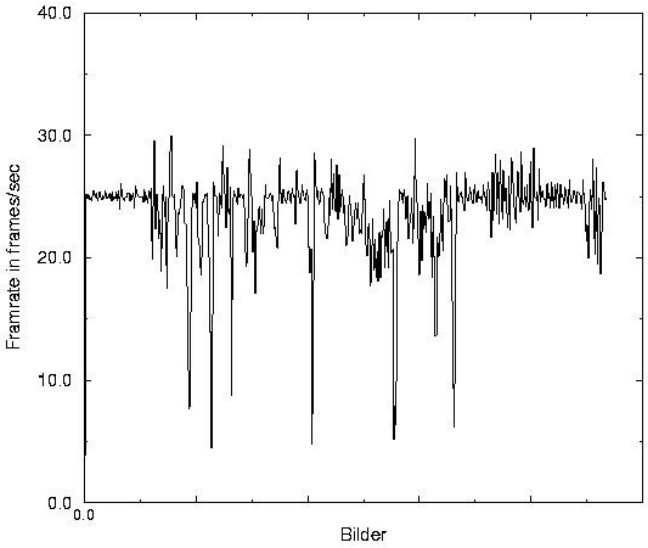



Abbildung 5.7: Framerate CVV (SGI-O2, 180 Mhz)

Dimension des Markers im Bild, womit Translationen und langsame Rotation recht gut erfasst werden können. Schnelleren Drehungen oder Richtungsänderungen bewirken jedoch eine größere Verschiebung im Bild, so dass gegebenenfalls die Marker den Suchbereich verlassen haben und daher eine neue Initialisierung durchgeführt werden muss. In diesem Fall muss das gesamte Bild wieder verarbeitet werden, wodurch eine deutliche Verzögerung eintritt.
Diese Zeitverzögerungen werden in Abbildung 5.7 veranschaulicht. Die Framerate des Trackingsystems wurde hierfür mit über 1000 Bildern (40 Sekunden) aufgezeichnet. Anhand kurzer Einbrüche mit einer Framerate von nur 5 bis 10 Hz können leicht die Zeitpunkte erkannt werden, in denen eine neue Initialisierung benötigt wurde. Mit dem CVV-System gewonnenen Erfahrungen und der Festlegungen seiner Schwachstellen wurde das VBT-Systems, das im folgenden Abschnitt präsentiert wird, entwickelt.

## 5.2 Das VBT-System

### 5.2.1 Vorgehensweise

Mit der Prototypisierung des optischen Trackers CVV wurde festgestellt, dass optische Trackingsysteme entsprechend ihres Einsatzgebietes sehr flexibel gestalten werden müssen. Je nach Anwendung sind beispielsweise einzelne Marker-Typen mehr oder weniger geeignet, oder es sind nur 2D-Bildtransformationen und nicht 3D-Lokalisierungsdaten notwendig. Ein weiterer wichtiger Aspekt besteht darin, dass bei der Bilderkennung nicht nur eine absolute Transformation zu einem globalen Koordinatensystem erforderlich ist, sondern gleichzeitig mehrere Transformationen bezüglich verschiedener unabhängiger Objekte benötigt werden. Dies ist beispielsweise der Fall, wenn zusätzlich zur Benutzerlokalisierung noch einen “Pointer” im Raum erfasst werden soll.
Letztlich sollte das System auch zu anderen Trackingsysteme offen sein. Die Kameraposition kann beispielsweise über einen externen Sensor, z.B. GPS oder Ultraschallsender, erfasst

werden, wobei mit Hilfe des Systems nur eine Verfeinerung im Bild für die Überlagerung erzielt werden soll. Um für diesen und ähnliche Fälle verschiedene Bildverarbeitungs- und Lokalisierungsverfahren leicht integrieren zu können, muss das System modular gestaltet sein.

### 5.2.2 Komponenten

Beim optischen Tracking können zwei grundlegende Module definiert werden. Mit Hilfe des ersten Moduls werden die benötigten Daten aus den gewählten Bildern extrahiert und verarbeitet. Das zweite Modul dient der Auswertung dieser Bilder zur Bestimmung geometrischer Transformationen.

#### Extraktion der Eingabedaten

Ziel dieses Moduls ist, die Bilder zu verarbeiten, vordefinierte Primitive zu extrahieren und diese zuletzt eindeutig identifizieren. Dafür werden die drei folgenden Komponenten definiert:

1. **Bildverarbeitung:** Die Komponente *Bildverarbeitung* dient der Erzeugung der Bildmerkmale. Hierbei werden Bildregionen, Konturverläufe, Kanten, Ecken oder Linien aus dem Bild extrahiert. Auf Grund der großen Menge an zu verarbeitenden Daten nimmt diese Komponente oft die meisten Rechnerzeit in Anspruch.

2. **Formanalyse (Shape-Analysis):** Durch die Komponente *Formanalyse* werden die Bildmerkmale zur Entwicklung einer komplexeren Struktur analysiert und gruppiert. An dieser Stelle werden beispielsweise aus den gefundenen Konturen Quadrate gebildet.

3. **Identifikation:** Mit Hilfe der Komponente *Identifikation* werden die extrahierten Bildstrukturen eindeutig identifiziert und gegebenenfalls den entsprechenden 3D-Informationen zugeordnet. Die Identifizierung kann u.a. über eine Codierung oder die Erkennung einer geometrischen Einordnung realisiert werden.

#### Bestimmung der geometrischen Transformation

Durch das Modul "Bestimmung der geometrischen Transformation" werden die geometrischen Transformationen berechnet. Dabei handelt es sich nicht unbedingt nur um die 3D-Kameraorientierung sondern auch um die 2D-planaren Transformationen in der Bildebene. Das Modul ist in folgende Komponenten untergeteilt:

1. **Analyse der 3D-Konfiguration:** Entsprechend der gefundenen Bilddaten wird mit Hilfe der Analyse der 3D-Konfiguration der passende Algorithmus ausgewählt. Wenn beispielsweise alle Marker in einer Ebene liegen, muss durch den Algorithmus diese Konfiguration explizit berücksichtigt werden.

2. **Bestimmung der geometrischen Transformation:** Diese Komponente bewirkt in erster Linie den Aufruf des ausgewählten Algorithmus zur Bestimmung der Kameraposition und -orientierung.

3. **Fehleranalyse:** Durch die Fehleranalyse wird die Gültigkeit der Transformation bestätigt, indem die numerische Stabilität und die Fehler der Berechnungen untersucht werden. Eine einfache Kontrollmöglichkeit ist z.B. durch die Berechnung der Fehler im Bild erzielbar.

4. **Rückprojektion:** Mit dem abschließenden Modul "Rückprojektion" können Verfeinerungen der vorhandenen Transformationsparameter vorgenommen werden. Auf Basis der ersten Werte der Transformationsparameter wird im Bild gezielt nach gegebenen Primitiven (Ecken, Punkten, Kanten) gesucht, und anschließend werden mit Hilfe der gefundenen Daten die Parameter nachjustiert. Dies bietet den Vorteil, dass nicht eindeutig identifizierbare Merkmale einbezogen werden können. Auf diese Weise können Marker mit und ohne ID in die Szene eingebracht werden.

## 5.3 Das VBT I-System

Mit der Realisierung von VBT I soll ein flexibles optisches Trackingsystem realisiert werden, das auf kleinen mobilen Rechner funktioniert und Szenarien wie beispielsweise "Augmented Reality für Service und Wartung" umsetzen kann.

### 5.3.1 Anforderungen

An das VBT I-System wird neben den üblichen hohen Genauigkeitsanforderungen die Anforderung gestellt, hinsichtlich einer hohen Anzahl unterschiedlicher Marker flexibel einsetzbar zu sein. Das System soll auf kleinen Computern, wie beispielsweise mit USB oder FireWire-Kameras ausgerüsteten Laptops oder Wearables, appliziert werden können und 3D-Tracking ermöglichen.

### 5.3.2 Markerdesign

Da eine leichte Bereitstellung des Trackingsystems eine große Rolle für die Akzeptanz dieser Technologie spielt, sollen einerseits die Marker leicht zu erstellen sein und anderseits das Tracking ohne aufwändige Vorbereitung, d.h. mit so wenig Markern wie möglich, in Betrieb genommen werden können.

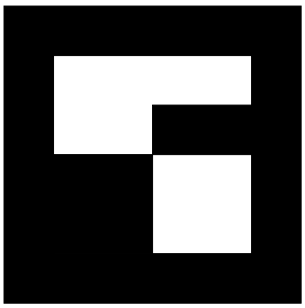


Abbildung 5.8: Beispiel eines kodierten Markers

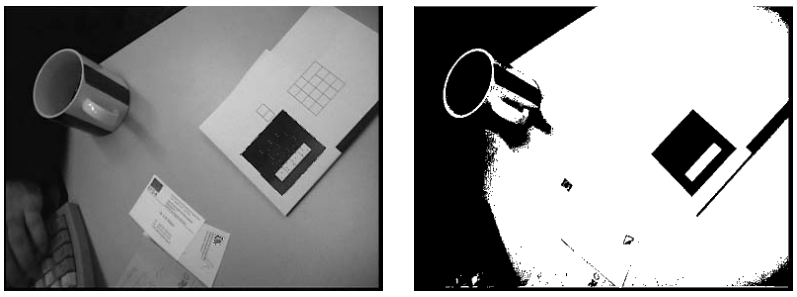
Eine quadratische Form des Markers erweist sich als besonders günstig, da in diesem Fall mit nur einem Marker die 3D-Lokalisierung der Kamera ermöglich wird. Für die Codierung

ist dabei ein $N \times N$ Feld in der Mitte des Quadrats vorgesehen, wobei für die meisten Anwendungen $N = 4$ angenommen wird.
Darüber hinaus sind dieser Marker schwarz auf einem weißen Hintergrund und können damit mit herkömmlichen Druckern erstellt werden. Ein Beispiel eines codierten Markers wird in Abbildung 5.3.2 veranschaulicht.

### 5.3.3 Extraktion der Marker

#### Segmentierung anhand eines einzigen Schwellwertes

Die Marker besitzen einen hohen Kontrast und können unter ausreichenden Lichtbedingungen durch eine einfache Binarisierung mit einem einzigen Schwellwert extrahiert werden. Das Hauptargument, das für die Wahl dieser einfachen Bildverarbeitung spricht, ist der minimale Rechnerzeitanspruch pro Pixel. Die Bestimmung der Schwellwerte stellt den kritischen Punkt des Verfahrens dar, da dieser für alle Bilder und über die ganze Bildfläche festgelegt ist.
Eine klassische Methode zur Schwellwertermittlung besteht aus der Analyse des Bildhistogramms und der Suche nach dem optimalen Trennwert der Pixel in zwei Klassen [67, 38]. Ein weiteres und schnelleres Verfahren stellt das sogenannte Min-Max-Verfahren dar. Dabei wird zuerst nach dem maximalen und minimalem Pixelwert gesucht und daraus der Mittelwert als Binarisierungsschwellwert ermittelt. Beide Verfahren sind jedoch nicht für Echtzeit-Anwendungen geeignet.



(a) (b)

Abbildung 5.9: Binarisierungsergebnisse

Da die Marker jedoch schwarze Objekte auf einem weißen Hintergrund darstellen, kann der Binarisierungsschwellwert einfach auf die Hälfte der idealen Pixelwerte der Marker und deren Hintergrund, auf den Pixelwert 127, festgelegt werden. Aus dieser Vereinfachung resultiert eine schnellere, echtzeitgeeignete Verarbeitung. Ein Beispiel einer Bildsegmentierung in dieser Art wird in Abbildung 5.9 gegeben.

#### Segmentierung mit lokalen, adaptiven Schwellwerten

Im Fall starker Schatten oder Lichtschwankungen erweist sich eine einfache Binarisierung jedoch als nicht ausreichend, siehe Abbildung 5.11. Die Marker werden nicht erfasst oder Kanten können nicht präzis lokalisiert werden. Eine Lösung liefert eine Binarisierung mit einem adaptiven Schwellwert, wobei dieser für jedes Pixel neu festgelegt werden muss.
Bei der adaptiven Min-Max-Methode wird eine quadratische Region um das Pixel untersucht und der lokale Schwellwert bestimmt, für alle Pixel in $(i, j)$ ist folgender Schwellwert

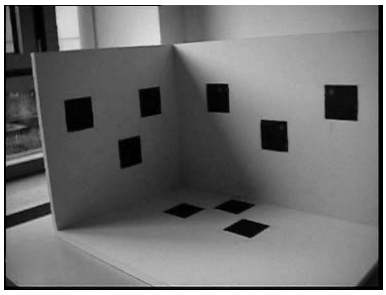


Abbildung 5.10: Ungleichmäßige Beleuchtung

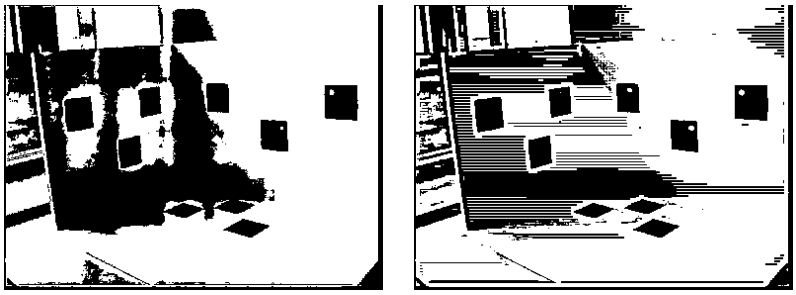
definiert:

$$T(i,j) = (P_{low} + P_{high})/2 \tag{5.3}$$

mit $P_{low}$ und $P_{high}$ als jeweils kleinstem bzw. größtem Pixelwert in einer Region $(W_i, H_j)$. Diese Region muss ausreichend Kontrast aufweisen, um die Festlegung eines neuen Schwellwerts zu rechtfertigen. Das bedeutet, dass die Differenz $(P_{high} - P_{low})$ eine minimale Größe überschreiten muss:

$$P_{high} - P_{low} > Kontrast \tag{5.4}$$

Der neue Schwellwert kann als solcher eingesetzt werden, siehe Abbildung 5.11(a), oder er kann dazu dienen, den globalen Schwellwert zu korrigieren, siehe Abbildung 5.11(b). Sogar unter extremen Bedingungen, wie z.B. starken Schatten oder Überbelichtung, werden mit dieser Methode die Quadrate richtig erfasst.



(a) (b)

Abbildung 5.11: lokale Korrektur durch Bildung eines adaptiven schwellwertes mit Hilfe eines globalen Schwellwertes (a) und durch lokale Min-Max-Methode (b)

Die beschriebenen Verfahren besitzen eine hohe Robustheit und eignen sich daher besonders für Anwendungen, bei denen nur zeitweise ein neues Bild ("Snapshot") benötigt wird. Für Echtzeit-Anwendungen ohne spezielle Hardware-Unterstützung sind sie jedoch auf Grund des hohen Rechenaufwandes wenig geeignet.

**Formanalyse**

Nach der Segmentierung werden die äußeren Konturen der Bildregionen erfasst und analysiert. Dabei soll eine viereckige Form erkannt und die Kanten oder Eckpunkten lokalisiert werden, siehe Abbildung 5.12.

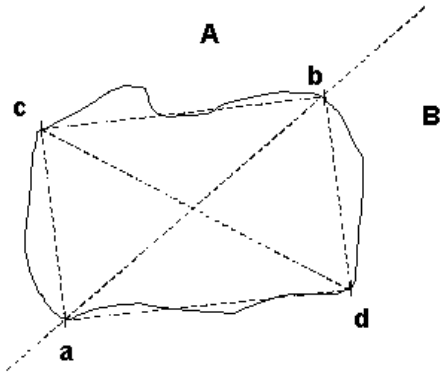



Abbildung 5.12: Bestimmung des Viereckes aus der Kontur einer Bildregion

Zuerst wird nach einer der Hauptdiagonalen des Kontursverlaufs gesucht. Sie wird als die Linie der entferntesten Konturpunkte definiert. Die Diagonale trennt die Kontur in zwei Teile $A$ und $B$ und legt die Punkte $a$ und $b$ fest. Anschließend werden die Punkte mit dem größten Abstand zur Diagonalen von jeweils $A$ und $B$ bestimmt. Sie bilden die zwei weiteren Ecken $c$ und $d$ des potentiellen Quadrats. Die Linien werden über die vier Kanten des Quadrats interpoliert und ihre Schnittpunkte als die vier Ecken des Markers angesetzt. Mit Hilfe heuristischer Schwellwerte über Kantenlänge und Interpolationsfehler wird ermöglicht, die quadratischen Formen herauszufiltern.

### Identifikation und Zuordnung

Die Identifikation beruht auf der Codierung der Marker. Alle Codierungsfelder werden zuerst abgetastet und die zugehörigen Werte gespeichert (Abbildung 5.13). Um die Merkmale des Codefeldes trotz perspektivischer Verzerrung richtig erfassen zu können, wird die Homographie $\boldsymbol{H}$ zwischen den Bildpunkten und den 3D-Koordinaten des Markers angewendet. Die Position der Abtastpunkte ist im Modell festgelegt und wird mit Hilfe von $\boldsymbol{H}$ auf die Bildebene transformiert.

Abbildung 5.13: Abtasten des Codierungsfeldes

Anschließend erfolgt eine Korrelation zwischen den Bildmerkmalen und allen für die Anwendung vorhandenen Markern. Den Bildmerkmalen werden für jede der vier möglichen Markerausrichtungen - pro Markerkandidat existieren vier Korrelationswerte - zugewiesen. Alle Ergebnisse werden in einer Matrix festgehalten.

| | Marker 1004 | Marker 108 | Marker 1005 | Marker 4231 |
|---|---|---|---|---|
| Kand. A | 0.12 | 0.82 | 0.31 | 0.41 |
| Kand. B | 0.76 | 0.21 | 0.77 | 0.14 |
| Kand. C | 0.50 | 0.37 | 0.71 | 0.21 |

Tabelle 5.1: Zuordnungsmatrix

Tabelle 5.1 stellt eine solche Matrix dar. In den Spalten werden alle potentiellen 3D-Marker und in den Zeilen alle im Bild gefundenen Markerkandidaten eingetragen. Die Korrelationswerte stellen die Matrixelemente dar.

**Zuordnung:** Auf Grund von Ungenauigkeiten und Fehlern bei dem Abtasten kann nicht davon ausgegangen werden, dass die Zuordnung immer eindeutig ist. Daher kann beispielsweise in Tabelle 5.1 nicht einfach nach dem besten Score jeder Zeile gesucht werden. In diesem Fall ergäbe sich für den Kandidat B die Zuordnung zu Marker 1005 mit Korrelationswert 0.77 und Marker C würde Marker 1004 mit dem Korrelationswert 0.5 zugewiesen. Die umgegekehrte Zuordnung ist jedoch wahrscheinlicher, da sie den Koeffizienten 0.76 und 0.71 entsprechen und höhere Scoresumme ($0.76 + 0.71 = 1,47$ statt $0.5 + 0.77 = 1,27$) liefern würde.
Die maximale Scoresumme liefert die höchste Wahrscheinlichkeit der richtigen Merkmalszuordnung, d.h. von allen möglichen Kombinationen wird die Zuordnung, die die Summe über alle Korrelationswerte maximiert, beibehalten. Alle Kombinationen auszuprobieren, erweist sich als äußerst aufwendig, da die Komplexität mit der Fakultät der Markeranzahl ansteigt. Für $n$ Marker sind $n!$ Kombinationen möglich.
Eine optimierte Lösung für Zuordnungsprobleme bietet die sogenannte ungarische Methode an [12]. Dieses Verfahren behandelt jedoch nur quadratische Matrizen und muss für Matrizen beliebige Dimensionen erweitert werden. Bei der Erweiterung werden zusätzliche Zeilen oder Spalten mit der Null-Koeffiziente addiert.

### 5.3.4 Ermittlung der Kameratransformation

#### Konfigurationsanalyse

Mit Hilfe der Komponente “Konfigurationsanalyse” wird die räumliche Konfiguration der Marker ausgewertet. Bei der Analyse wird überprüft, ob die Marker in einer Ebene liegen. Gegebenfalls wird die Transformationsmatrix zu der Ebene $z = 0$ bestimmt und die Marker in diese Ebene transformiert.

#### Ermittlung der Kameratransformation

Bei der Ermittlung der Kameratransformation wird die Position und Orientierung der Kamera aus der 2D-Homographie zwischen den Markern im Bild und den Markern in der realen Welt berechnet. Hierdurch wird die Planaritätseigenschaft der Marker berücksichtigt und es können somit stabilere Ergebnisse erzielt werden.
Im folgenden wird die Ableitung dieser Kameratransformation aus der Homographie erläutert. $\boldsymbol{M}(X, Y, Z, 1)$ sei eine Markerecke und $\boldsymbol{m}(x, y, 1)$ ihre Projektion im Bild. Dann wird der Punkt $\boldsymbol{M}$ in $\boldsymbol{m}$ über die Homographie $\boldsymbol{H}$ wie folgt transformiert:

$$s\boldsymbol{m} = \boldsymbol{H}\boldsymbol{M} \tag{5.5}$$

wobei $s$ einen beliebigen Skalar darstellt. Mindestens vier Punktpaare $(\boldsymbol{m}, \boldsymbol{M})$ werden für die Berechnung von $\boldsymbol{H}$ benötigt. Die Gleichung (5.5) führt zu einem Linearsystem, das mit Hilfe eines SVD- Verfahrens (Singular Value Decomposition) gelöst werden kann [69].
Wenn die Punkte $\boldsymbol{M}$ in der Ebene $Z = 0$ liegen, resultieren daraus folgende Kameragleichungen, siehe Kapitel 3:

$$s\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \boldsymbol{A}(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{t})\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \boldsymbol{A}(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{t})\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{5.6}$$

und aus der Gleichung 5.5 erhält man:

$$\boldsymbol{H} = \boldsymbol{A}(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{t}) \tag{5.7}$$

Da die internen Parameter der Kamera bekannt sind, kann die Transformation $(\boldsymbol{R}, \boldsymbol{t})$ direkt abgeleitet werden:

$$\boldsymbol{H}' = \boldsymbol{A}^{-1}\boldsymbol{H} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{t}) \tag{5.8}$$

bzw.

$$(\boldsymbol{h}'_1, \boldsymbol{h}'_2, \boldsymbol{h}'_3) = s(\boldsymbol{r}_1, \boldsymbol{r}_2\boldsymbol{t}) \tag{5.9}$$

Die zwei Vektoren $\boldsymbol{r}_1$ und $\boldsymbol{r}_2$ der Rotationsmatrix $\boldsymbol{R}$ sind folgenderweise bestimmt: $\boldsymbol{r}_1 = s\boldsymbol{h}'_1$, $\boldsymbol{r}_2 = s\boldsymbol{h}'_2$. Der dritte Vektor steht orthogonal zu $\boldsymbol{r}_1$ und $\boldsymbol{r}_2$ und wird mit $\boldsymbol{r}_3 = \boldsymbol{r}_1 \wedge \boldsymbol{r}_2$ definiert. Der Translationsvektor $\boldsymbol{t}$ ist durch $\boldsymbol{t} = s\boldsymbol{h}'_3$ mit dem Skalarfaktor $s = 1/\|\boldsymbol{h}'_1\| = 1/\|\boldsymbol{h}'_2\|$ gegeben.
Wenn die Ergebnisse nicht genau genug sind, siehe Abschnitt "Fehleranalyse", wird anschließend eine Verfeinerung der Parameter mit Hilfe eines nicht-linearen Optimierungsverfahrens vorgenommen. Das Verfahren wird im Kapitel 4 detailliert beschrieben und diskutiert.

### Fehleranalyse

Die Fehleranalyse kontrolliert den Fehlermittelwert aller Marker. Dafür wird hier der sogenannte "Root Mean Square" $rms$ verwendet.

$$rms = \sqrt{\frac{1}{N}\sum_{1}^{N}(\boldsymbol{m}_i - \boldsymbol{P}\boldsymbol{M}_i)^2} \tag{5.10}$$

wobei $\boldsymbol{P}$ die Projektionsmatrix der Kamera, $\boldsymbol{m}_i$ und $\boldsymbol{M}_i$ die Markereckpunkte jeweils im Bild bzw. in 3D sind.

#### Rückprojektion

Die Rückprojektion stellt einen wesentlichen Schritt der 2D/3D-Bildanalyse dar. Ihr Vorteil besteht darin, dass die örtlichen Informationen der gesuchten Bildmerkmale schon bekannt sind und dass gezielte Verfeinerungen vorgenommen werden können, siehe auch Abschnitt 5.1.7. Die Rückprojektion erweist sich auch als einfache Möglichkeit, neben den kodierten Markern zusätzliche Merkmale für das Tracking zu berücksichtigen, siehe auch Abschnitt 5.1.4 "CVV". Ein wichtiger Operator stellt die Bestimmung von Ecken dar. Somit können nicht nur die Marker neu erfasst werden, sondern auch alle weiteren natürlichen und kontrastreichen Ecken der Szene.
Für die Operation wurde der Harris-Operator ausgewählt. Untersuchungen in [14] zeigen, dass dieser Operator die besten Leistungen gegenüber Genauigkeit, Robustheit und Bildrauschen liefert.

## 5.4 Das VBT-II-System

### 5.4.1 Farbige Marker

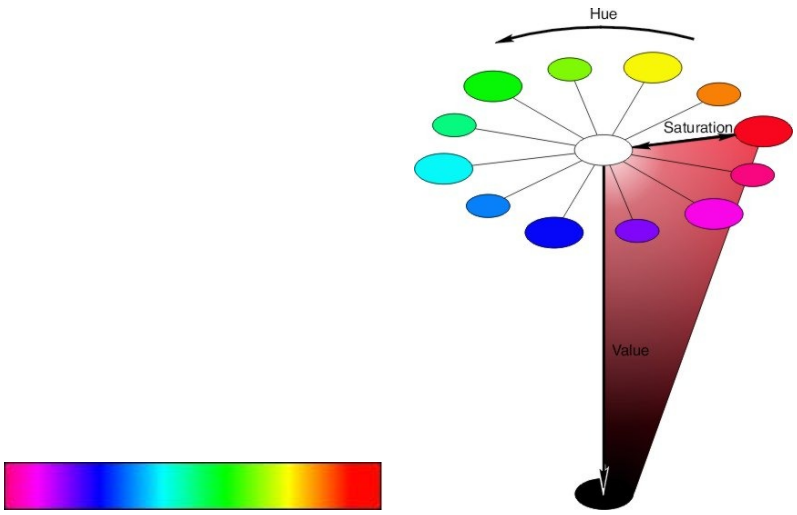
Bisher wurden farbige Marker für Photogrammetrie-Aufgaben oder Kamera-Tracking nur selten angewendet. Farbige Marker beinhalten jedoch zusätzliche Informationen, deren Berücksichtigung eine schnellere und robustere Detektion ermöglichen kann. Dieser Ansatz wird als eine Verknüpfung zwischen Verfahren, die auf passiven schwarz-weiß Markern basieren, und Verfahren, die aktive Marker (LEDs oder Infrarotlicht) nutzen, betrachtet.
In [18] werden verschiedene Untersuchungen über Farbsegmentierung im RGB-Raum vorgestellt und anschließend ein Trackingsystem mit farbigen Markern vorgestellt. Um die Farbe zu klassifizieren, wird bei dem Verfahren der Winkel zwischen den RGB-Farbvektoren genutzt. Dennoch besteht im RGB-Raum die Schwierigkeit, eine von der Intensität unabhängige Darstellung zu bestimmen und das Farbvolumen des Markers über einfache Schwellwerte zu erfassen. Eine günstigere Darstellung der Farbwerte des Pixels ist durch das HSI-Modell gegeben [87].

### 5.4.2 Extraktion

#### Segmentierung im HSI-Farbraum

**Das HSI-Farbsystem**
Die Farbdarstellung erfolgt über den Farbwert H (Hue), die Sättigung S (Saturation) und die Helligkeit I (Intensity):

- **Hue:** Hue bezeichnet die dominante Wellenlänge der Farbe. Die Farbwellenlängen werden über einen Kreis dargestellt, d.h. von 0 bis 360 Grad. Hue wird über den Winkel zum Ursprung charakterisiert.
- **Saturation:** Die Sättigung gibt die Entfernung der Farbe von Grau in einem Bereich von 0% bis 100% an. Eine Sättigung von Null entspricht der Farbe Weiß.
- **Intensity (oder Lightness):** Die Intensität erfasst die gesamte Helligkeit des Pixels auf einer Skala (0% bis 100%). Daraus folgt, dass das HSI-System über einen Kegel dargestellt werden kann, siehe Abbildung 5.16(b).




(a) (b)

Abbildung 5.14: Spectrum (a), Hue(R)=0 deg; Hue(G)=120 deg; Hue(B)=240 deg (b)


Abbildung 5.15: Sättigungsskala der Farbe Magenta

Dank dieses Ansatzes können die Marker eine beliebige Farbe besitzen. Dabei sollte dennoch beachtet werden, dass Marker mit einer hellen (hohe Intensität) und gesättigten Farbe in Videobildern leichter zu finden und daher besser für das Tracking geeignet sind.

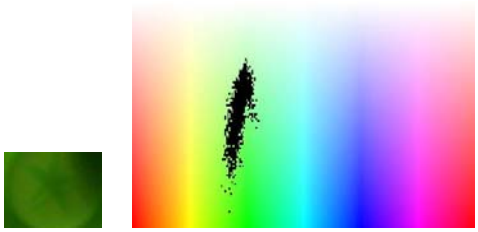
**Konvertierung vom RGB- zum HSI-Farbraum**

Die Videobilder werden im RGB-Modell eingelesen und müssen zuerst in das HSI-Modell konvertiert werden.

**Charakterisierung und Segmentierung**

Die Farbe des gesuchten Objektes wird über ein Muster, das aus einem Bildauszug besteht, charakterisiert. Die hier relevanten Komponenten sind, da eine Unabhängigkeit von der Intensität bestrebt wird, der Farbwert (H) und die Sättigung (S).

In Abbildung 5.16 wurden beispielsweise die Hue- und Saturation-Werte eines grünen Markers auf dem H-S-Farbspektrum eingetragen.



(a) (b)

Abbildung 5.16: (a) Objektfarbe und (b) Pixeldarstellung in H und S Ebene

Die Lokalisierung des Musters auf der Hue-Saturation-Ebene ist eindeutig und kann gut

mit wenigen Schwellwerten erfasst werden.
Mit einer Farbe, die nicht oft in der Szene vorkommt, genügen diese Informationen, um der Marker zu extrahieren. Zwei weitere Beispiele stellt Abbildung 5.17 dar.

(a) (b)

Abbildung 5.17: (a) Originalbild, (b) Ergebnisse der Farbdetektion

Im ersten Fall wurde beispielsweise eine grüne Münzmarke aufgenommen, siehe erste Zeile links, und mittels zweier Schwellwerte für die Farbwerte und einem niedrigen Schwellwert für die Sättigung detektiert.
Wie auch anhand der Bilder in der zweiten und dritten Zeile verdeutlich wird, können mit dieser Methode Farben leicht erfasst werden. Da der Hue-Wert unabhängig von der Intensität ist, wird eine zuverlässige Farbsegmentierung auch unter heterogenen Beleuchtungsbedingungen ermöglicht.
Folgende Einschränkungen müssen jedoch beachten werden:

1. Die Größe H (Hue) ist extrem rauschenempfindlich und dadurch wenn die Intensität I sehr niedrig bzw. sehr hoch ist, schlecht definiert.

2. Die Größe H (Hue) ist auch schlecht definiert, wenn die Sättigung S sehr niedrig ist.

3. Die Sättigung S ist schlecht definiert, wenn die Intensität I sehr niedrig oder sehr hoch ist.

Wenn die oben erwähnten Fällen auftreten, erfolgt die Segmentierung nur auf Basis der Intensität des Markers im Bild.

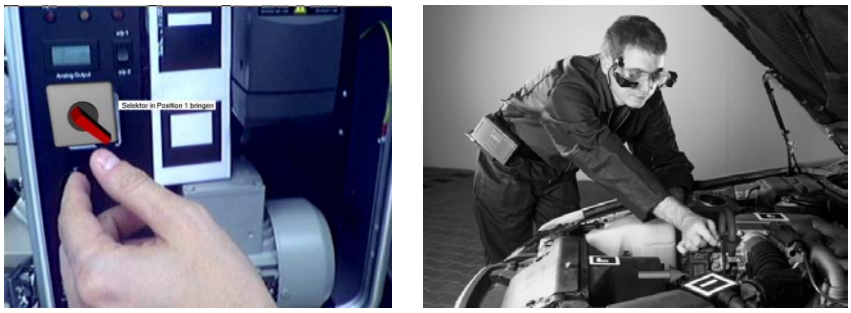
#### Scanning und Region-Growing

Da allein über die Farbanalyse potentielle Marker zuverlässig gefunden werden können, besteht die Möglichkeit, die Bildverarbeitung durch Verwendung farbiger Marker an Stelle von schwarz-weißen Markern effizienter zu gestalten. Im weiteren wird eine mit einem *Region Growing*-Verfahren kombinierte Scanning-Funktion dafür eingeführt.
Bei dieser Vorgehensweise wird, an Stelle einer kompletten Bildbearbeitung, nur jede $n$-Spalte und $p$-Linie gelesen, wobei die Parameter $n$ und $p$ in Abhängigkeit zu den minimalen Dimensionen der Marker im Bild stehen. Wenn ein Pixel erkannt wird, folgt direkt eine Region-Growing-Funktion, um das komplette Objekt zu erfasst. Gleichzeitig werden die geometrischen Eigenschaften der Region (Mittelpunkt, Fläche und Momente) rekursiv berechnet.
Die Kombination *Scanning* und *Region-Growing* ermöglicht, das Bild schnell und mit geringem Rechneraufwand zu analysieren.

## 5.5 Anwendungen

### 5.5.1 Das ARVIKA-Projekt

Die effiziente Informationsvermittlung bei beispielsweise der Ausbildung von Service-Technikern oder/und der Dokumentation von Arbeitsprozessen wird heutzutage für viele Unternehmen auf Grund immer komplexerer Produkte zunehmend schwieriger. Diese Problematik betrifft insbesondere Anwendungsbereiche wie Automobil- und Flugzeugbau, Maschinen- und Anlagenbau und erfordert verbesserte Diagnose- und Informationssysteme.




(a) (b)

Abbildung 5.18: Anwendungsszenario aus ARVIKA: Wartung einer Maschinesteuerung (a) und Reparaturvorgang in einer Automobil-Werkstatt (b)

Augmented Reality kann dabei ein situationsgerechtes Agieren unterstützen. So können vor Ort und beispielsweise während der Ausführung einer Wartungsaufgabe die benötigten Informationen direkt am realen Objekt eingeblendet werden, siehe Abbildung 5.18. Dadurch ist zum Beispiel möglich einem Service-Mitarbeiter die einzelnen Arbeitsschritte im Sinne eines Workflows in visueller und sofort verständlicher Form zu präsentieren.
Um die Entwicklung und den Einsatz von AR-Anwendungen zu fördern, haben sich zwanzig Unternehmen und Forschungseinrichtungen im Projekt “ARVIKA” (www.arvika.de) zum weltweit größten AR-Konsortium zusammengeschlossen. Die entwickelten Prototypen und Anwendungen basieren auf ddem präsentierten Trackingsystem VBT I.

### 5.5.2 Die Cybernarium-Days

Das Fraunhofer-IGD veranstaltete im Jahr 2002 eine Ausstellung von AR- und VR-Anwendungen mit dem Ziel, das didaktische Potential dieser Technologie zu demonstrieren. Die sogenannten *Cybernarium Days* öffneten einen Vorausblick auf den geplanten Science-Park *Cybernarium* und wurden von rund 10.000 Besucher besucht. Insgesamt 15 Exponate, gruppiert um die Themen Lernen, Spiel, Arbeit und Kunst wurden präsentiert. Darunter wurde die beide Spiele "Augmented Tic-Tac-Toe" und das "Augmented-Memory"-Spiel und die Demonstration "dursichtige Patient" vorgestellt. Bei diesen Präsentationen wurde für die Kameralokalisierung das VBT-I-System angewendet.

## 5.6 Zusammenfassung

In diesem Kapitel wurde gezeigt, dass auf Basis von optischen Methoden möglich ist, die Tracking-Anforderungen von AR zu erfüllen. Die Kamera hat sich als ein präzises, leicht-einsetzbares und mobiles Trackinggerät erwiesen.
Drei auf komplementären Markern basierende Verfahren wurden hierfür entwickelt. Das erste System, CVV, basiert auf einer iterativen "Punkt-Linien"-Distanz-Minimierungsmethode, die besonders gegen Teilverdeckungen robust ist und hohe Echtzeit-Performance anbietet. Auf Grund des iterativen Verfahrens und der lokalen Suche der Bildmerkmale müssen die Kamerabewegungen bei diesem Verfahren kontinuierlich und moderat sein. Im Gegensatz dazu stellen für die beiden Verfahren VBT I und VBT II starke und ruckartige Bewegungen keine Einschränkungen dar. Beide Methoden sind für HMD-Anwendungen sehr geeignet. Die hierbei eingeführten farbigen Markern ermöglichen eine zuverlässige und lichtunabhängige Bildsegmentierung im HSV-Raum. Die Bildbearbeitung konnte durch den Einsatz der farbigen Marker optimiert und eine sehr hohe Leistung, auch auf kleinen Rechnern, erreicht werden.

# Kapitel 6

# Markerloses optisches Tracking

In diesem Kapitel werden Verfahren des markerlosen optischen Trackings behandelt. Im ersten Abschnitt wird ein Überblick über die Thematik gegeben und das Grundprinzip der Stützung im Tracking definiert. Anschließend wird ein neues Konzept für ein optisches Trackingsystem ohne Marker vorgeschlagen, das auf sogenannten Referenzbildern und Bildregistrierungsverfahren basiert. Zahlreiche Untersuchungen und die Umsetzung eines ersten Prototyps zeigen die Machbarkeit und das Potential dieses Ansatzes. Das Tracking-System funktioniert mit einer Wiederholrate von 15 Hz auf einem Labtop und wurde im Rahmen von Mobil- und Outdoor-Szenarien getestet.

## 6.1 Markerloses Tracking

### 6.1.1 Problemstellung

Wie im Kapitel 5 “Markerbasiertes, optisches Tracking” dargelegt wurde, bleibt der Einsatz von Markern durch praktische und auch ästhetische Faktoren begrenzt. Markerloses Tracking kann von daher nicht nur als eine ideale und wünschenswerte Lösung betrachtet werden, sondern es stellt auch eine notwendige Technologie dar, um AR für spezielle Anwendungsbereiche zu ermöglichen.
Die Zurückgewinnung der Kamerabewegungen und der Szenestruktur aus Bildern ist seit ungefähr 20 Jahren eines der Hauptforschungsthemen der Robotik, der künstlichen Intelligenz und der Computer Vision. Dabei hat sich die Lösung dieses Problems als eine sehr komplexe Aufgabe erwiesen.
Im ersten Arbeitsschritt muss die Struktur der Szene erfasst und anschließend vom Rechner richtig interpretiert werden, um darauf basierend präzise Rückschlüsse über die aktuelle Kameraposition und -orientierung im Raum treffen zu können. Jeder einzelne Schritt ist schwierig, da kein Standardverfahren angewendet werden kann, sondern problembezogen eine passende mathematische Methode gefunden werden muss. Allein die Gesetze des 3D-*Computer-Sehens* sind erst kürzlich verstanden und mathematisch beschrieben worden [32, 46].
Weiterhin kann auf Grund des breiten Anwendungsgebietes für AR nicht einfach auf Gemeinsamkeiten der Umgebungen, wie z.B. Straßenrändern für kamerageführte Fahrzeuge, zurückgegriffen werden. Das Verfahren muss allgemein anwendbar bleiben.
Eine weitere Schwierigkeit stellt die Art der zu verfolgenden Bewegungen dar. Da die Bewegungen rückartig und unvorsehbar sein können, ist die Annahme eines Bewegungsmodells

kaum möglich, wenn nicht sogar ausgeschlossen.
Die Umsetzung des markerlosen Trackings in eine robuste Implementierung erscheint in dieser Hinsicht als eine kaum lösbare Aufgabe. Eine Analyse der spezifischen Anforderungen von AR und die Auswertung von Besonderheiten der Bildaufnahme, wie z.B. Kopfbewegungen, Aktionsradius, liefern jedoch wie im nächsten Abschnitt präsentiert, erste, wertvolle Ansätze.

### 6.1.2 Stützungsansätze des markerlosen Trackings

#### Die Notwendigkeit einer Stützung

In diesem Abschnitt werden zunächst die grundsätzlichen Ansätze für markerloses Tracking vorgestellt und anschließend auf die Notwendigkeit einer Stützung für bildbasierte Trackingverfahren vom AR hingewiesen.
In den Robotik- und Computer-Visions-Bereichen sind viele Verfahren, die nach einem rein iterativen Prinzip arbeiten, zu finden. Markante Merkmale der Umgebung, wie z.B. Linien oder Ecken werden von Bild zu Bild verfolgt und daraus die 3D-Struktur und die Kamerabewegung abgeleitet. Die Annahmen sind hierbei, dass

1. die Kamerabewegungen gleichmäßig sind und

2. eine absolute Skalierung und der Bezug zu einem globalen, vordefinierten Koordinatensystem nicht notwendig oder vorgegeben ist.

Die Bewegungen in AR sind dagegen häufig rückartig und unvorhersehbar. Ein auf diesen Annahmen beruhendes Trackingverfahren würde beispielsweise bei einer starken Bewegung oder einer kurzen Sichtverdeckung der Kamera scheitern. Das System wäre nicht in der Lage, sich autonom zu reinitialisieren, da der Zusammenhang mit dem vorherigen Bild nicht mehr vorhanden ist. Um den Bezug zu den virtuellen Objekten korrekt erstellen zu können, werden absolute Positions- und Orientierungswerte benötigt. Eine sogenannte Stützung, die den Bezug zur absolute Koordinaten liefert, ist deswegen notwendig.
Bei einem Stützungsansatz sollen hier keine klassischen Marker, die den Bewegungsraum und die Anwendungsmöglichkeiten einschränken, verwendet werden, sondern die 3D-Position und -orientierung der Kamera soll allein anhand der Bilddaten abgeleitet werden.

#### Stützung durch Referenzbilder

Im folgenden wird ein Verfahren, das eine Stützung durch ein oder mehrere Bilder der Szene anbietet, vorgestellt und erläutert. Die zur Stützung verwendeten Bilder werden “Referenzbilder” genannt und sind vollkalibriert, das bedeutet Kameraposition und -orientierung sowie die internen Kameraparameter sind bekannt.
Beim Tracking wird das aktuelle Live-Videobild mit den Referenzbildern verglichen und das Bestpassende herausgefiltert (siehe Abbildung 6.1). Anschließend wird die Transformation zwischen dem Referenzbild und dem Live-Bild berechnet und daraus die neue Kameraorientierung abgeleitet.

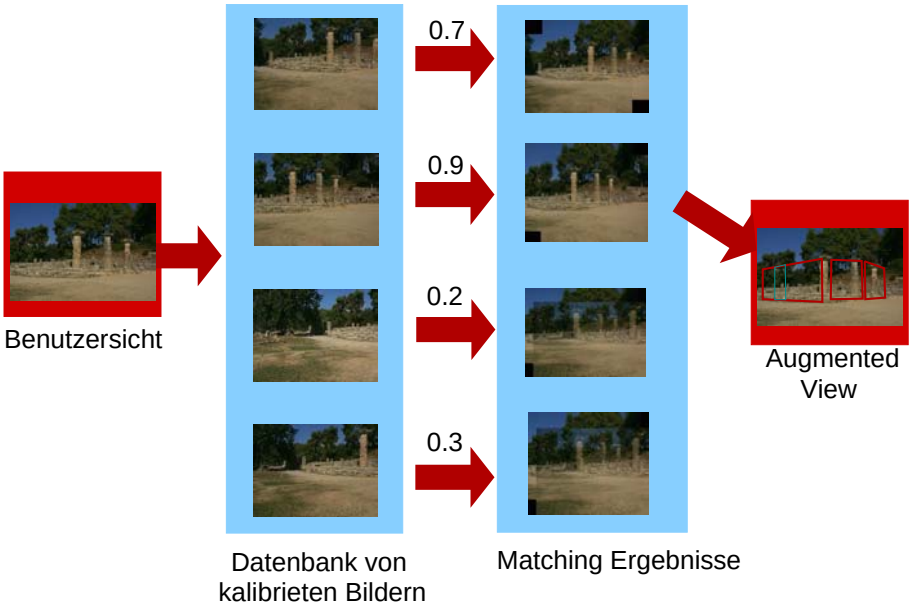



Abbildung 6.1: Tracking-Stützung durch Referenzbilder

Darüber hinaus werden zu jedem Referenzbild multimediale Daten, wie Ton, Text, Overlays und virtuelle Objekte assoziiert. Diese Informationen erfordern nicht immer 3D-Kenntnisse und können auch mit nur einem Bildbereich oder sogar einem Bildpunkt verknüpft werden. Die beiden Komponenten, Referenzbilder und multimedialen Informationen, bilden die *AR-Datenbank* des Systems.

### Vorteile der neuen Lösung

Das neue Verfahren bietet folgende Vorteile:

- **Einfache Umsetzung:** Das Trackingsystem beruht nur auf Standard- Bildern und benötigt außer einer Kamera keine weiteren Geräte. Die Umgebung bleibt unberührt.

- **Flexibilität:** Es wird weder ein Szenemodell noch eine Annahme über die Struktur der Szene benötigt. Die Vorbereitungszeit ist daher minimal.

- **Präzision:** Die virtuellen Informationen können in der realen Umgebung genau positioniert werden, da das System nach dem Inside-Out-Prinzip, siehe 2.3.1, arbeitet. Darüber hinaus gibt es kein Drift-Problem, weil die Transformation für jedes Bild neu berechnet wird und so Fehler nicht über die Zeit akkumuliert werden.

- **Robustheit:** Es wird keine Annahme über die Kamerabewegung getroffen. Abrupte Bewegungen sind erlaubt.

Diese neue Methode stellt zur Zeit erstmalig eine Methode dar, die auch für weiträumige Umgebungen (Out-Door) geeignet ist.

**Anforderungen**

**Vorbereitungsphase:** Für die Eingabe des Trackingsystems werden Bilder aus den relevanten Blickpunkten aufgenommen und als Referenzbilder gespeichert. Den Bildern werden die jeweiligen Blickpunktpositionen und -orientierungen zugewiesen.
Für ein AR-Szenario muss somit eine Reihe von Referenzbildern, die die Wissensbasis des Systems bilden, zur Verfügung stehen. Alle für diese Arbeit benötigten Methoden sowie die Ergebnisse ihrer Implementierung werden im Kapitel 4 beschrieben.

**Tracking vor Ort:** Für das Tracking vor Ort werden die Live-Videobilder mit den Referenzbildern verglichen und gegebenenfalls registriert. Den Kern des Trackingverfahrens stellt die Registrierungsmethode dar, die die zwei folgenden Anforderungen erfüllen muss:

- Bestimmung der Bildtransformation zwischen Referenz- und Live-Bild.
- Definition eines Gütekriteriums, das die Validität der Bildregistrierung widerspiegelt.

Als weitere Anforderung muss das Verfahren kleine Szeneänderungen oder neue Lichtverhältnisse zulassen, da, insbesondere für Out-Door-Applikationen, Veränderungen nicht ausgeschlossen werden können. In Abhängigkeit der Uhrzeit verändert sich beispielsweise die Stellung der Sonne, d.h. Beleuchtung und Schatten verändern sich. Auch ist beispielsweise das Betreten der Szene von Personen durchaus denkbar und eine wahrscheinliche Problemstellung.

## 6.2 Bildregistrierung

### 6.2.1 Einleitung

In diesem Abschnitt werden einleitend die theoretischen Grundlagen der Bildregistrierung erläutert. Anschließend erfolgt eine Zuweisung der Bildregistrierungsmethoden in die folgenden drei Klassen:

1. **Intensitätsbasierte Verfahren**
   Intensitätsbasierte Verfahren arbeiten direkt mit den Pixelwerten der Bilder und versprechen eine hohe Genauigkeit [82, 80].

2. **Merkmalbasierte Verfahren**
   Merkmalbasierte Verfahren extrahieren zunächst Merkmale aus den Bildern und finden dann Korrespondenzen zwischen den extrahierten Merkmalen [93].

3. **Frequenzbasierte Verfahren**
   Die frequenzbasierten Verfahren analysieren die Bildsignale im Frequenzraum beispielsweise mit Hilfe der Fouriertransformation [16, 73].

### 6.2.2 Definition der Bildregistrierung

Unter Bildregistrierung wird das deckungsgleiche Übereinanderlegen verschiedener digitaler Bilder verstanden. Im einfachsten Fall sollen zwei Bilder, ein Quellbild (Source) und ein

Zielbild (Target), aufeinander registriert werden. Die zu registrierenden Bilder müssen dabei teilweise überlappenden Inhalt haben, d.h. sie müssen einen Ausschnitt aus derselben Szene zeigen.
Bildregistrierungsverfahren versuchen eine optimale geometrische Transformation zu finden, mit der das Quellbild $I_1$ auf das Zielbild $I_2$ pixelgenau übereinandergelegt werden kann. In Abbildung 6.2 sind beispielsweise zwei Bilder mit einer projektiven Transformation registriert worden.

Abbildung 6.2: Beispiel einer Bildregistrierung

Mathematisch kann der Vorgang des Registrierens so definiert werden:

$$I_2(x', y') = I_1(f(x, y)) \tag{6.1}$$

Hierbei sei $f(x, y)$ eine Funktion, die die Koordinaten $(x, y)^\top$ aus dem Quellbild $I_1$ auf die Koordinaten $(x', y')^\top$ im Zielbild $I_2$ abbildet.
In einigen Anwendungsfällen kann auch eine zusätzliche Anpassung der Intensitätswerte notwendig sein. Dies ist z.B. dann der Fall, wenn die Bilder mit unterschiedlichen Sensoren aufgenommen worden sind, es stark reflektierende Objekte in der Szene gab oder Veränderungen der Lichtquelle eingetreten sind. Der Zusammenhang wird wie folgt beschrieben:

$$I_2(x', y') = g(I_1(x', y')) \tag{6.2}$$

Die Funktion $g$ sei eine photometrische oder radiometrische Transformation, die die Grauwerte der Bilder anpasst.
Eine photometrische Transformation arbeitet direkt mit den einzelnen Pixelwerten. Eine genaue Kenntnis der Szene ist hierbei nicht notwendig. Die lineare Grauwerttransformation ist ein Beispiel hierfür. Bei einer radiometrischen Transformation werden Eigenschaften der Szene, wie beispielsweise Reflektionseigenschaften der Objekte, Lichtquellen usw., zur Anpassung der Grauwerte betrachtet. Dazu muss die Szene ausreichend bekannt und modelliert sein.

### 6.2.3 Modell von Brown

Eine sehr umfangreiche Arbeit, die sich mit der Klassifizierung und Beschreibung von Bildregistrierungsmethoden befasst, wurde von Brown [13] veröffentlicht. Brown erfasst wesentliche Merkmale von Bildregistrierungsmethoden. Zusätzlich beschreibt sie ein Modell, dass allen Registrierungsverfahren zu Grunde liegt. So besteht jedes Verfahren aus den vier Komponenten:

1. Merkmalsraum
2. Vergleichsmetrik
3. Suchraum
4. Suchstrategie

Der *Merkmalsraum* besteht aus extrahierten Informationen, die für den Bildvergleich verwendet werden. Im Falle der intensitätsbasierten Verfahren handelt es sich dabei um die Intensitäts- oder Luminanzwerte der einzelnen Pixel. Bei merkmalbasierten Verfahren sind dies Merkmale[1], die aus den Bilddaten zunächst extrahiert werden müssen.
Die *Vergleichsmetrik* stellt ein Maß bereit, mit dem die gewonnenen Merkmale verglichen werden. Dies können z.B. Intensitätsunterschiede sein.
Durch *den Suchraum* werden die zulässigen geometrischen Transformationen beschrieben, mit denen die Bilder aufeinander registriert werden können, siehe Abschnitt 6.2.3.
Die *Suchstrategie* beschreibt das Verfahren, mit dem die nächste Iteration gefunden werden soll, um die optimale Transformation zu ermitteln.

#### Geometrische Transformation der Bilder

Die Kamerabewegung zwischen den Aufnahmen zweier Bilder spielt eine wichtige Rolle. Sie bestimmt die geometrische Transformation, mit der die Bilder registriert werden können. Bei einer allgemeinen Kamerabewegung[2] reicht eine globale geometrische Transformation nicht mehr aus, da sich einzelne Bildpunkte in den aufeinander folgenden Bildern unterschiedlich verschoben haben. In diesem Fall müssen lokale Transformationen gefunden werden. Die beschriebene Problematik wurde im Kapitel **??** vorgestellt und dort mit Hilfe eines Korrelationsoperators gelöst. Für Echtzeit-Tracking ist dieser Lösungsansatz jedoch auf Grund möglicher starker Bewegungen weniger geeignet.
Eine Ausnahme stellen spezielle Fälle, wie besondere Kamerabewegungen, dar. Sie sind für das markerlose Tracking interessant, da sie zuverlässig und in Echtzeit berechnet werden können. Ein besonderer Fall liegt beispielsweise vor, wenn die Kamera nur gedreht wurde. Die Transformation zwischen den Bildern kann hier als eine projektive Linear-Transformation (2D-Kollineation oder Homographie) erfasst werden.

[1] Häufig verwendete Merkmale sind Ecken, Kanten oder Konturverläufe.
[2] Dies ist z.B. der Fall, wenn der Kamerastandort geändert wird.

Mathematisch lässt sich die 2D-Transformation wie folgt beschreiben:
$\boldsymbol{m}$ und $\boldsymbol{m'}$ seien zwei Bildpunkte eines 3D-Punktes $\boldsymbol{M}$ im ersten und zweiten Bild. Wenn die Kamerabewegung zwischen beiden Bildern aus nur einer Rotation $\boldsymbol{R}$ besteht, gilt:

$$\begin{aligned} \boldsymbol{m} &= \boldsymbol{AIM} \\ \boldsymbol{m'} &= \boldsymbol{ARM} + \boldsymbol{t} \\ \boldsymbol{m'} &= \boldsymbol{ARA^{-1}m} \end{aligned} \tag{6.3}$$

Da die Matrizen $\boldsymbol{A}$ und $\boldsymbol{R}$ invertierbar sind, ist auch die Matrix $\boldsymbol{H} = \boldsymbol{ARA^{-1}}$ invertierbar und $\boldsymbol{H}$ eine Kollineation. Die Transformation wird auch Homographie genannt und stellt eine projektive Transformation dar. Die Matrix $\boldsymbol{H}$ ist bis zum Skalierungsfaktor definiert und hängt deswegen von acht Parametern ab.
Eine ähnliche Situation liegt für eine Szene, die aus einem planaren Objekt besteht, vor. Alle Bildpunkte beschreiben dieselbe lineare Bewegung. Durch die 2D-Kollineation wird die Transformation zwischen den Punkten im Raum und in der Bildebene nachgebildet.

**Spezielle Fälle und mögliche Annäherungen**

In der Praxis wird die Kamera am HMD getragen, d.h. dass die Kamerabewegungen hauptsächlich nur die Rotationskomponente beinhalten und dass, solange der Benutzer seine Position nicht verändert, Verschiebungen vernachlässig werden können.
In Tabelle 6.1 sind mögliche Kamerabewegungen und die entsprechenden Bildtransformationen aufgeführt. So können Bilder, bei denen die Kamera um die optische Achse gedreht und eventuell auch gezoomt wurde, mit einer euklidischen Transformation überlagert werden. Dies ist ebenso der Fall, wenn die Kamera um die optische Achse parallel verschoben oder eine geringe Drehung um die Stativachse durchgeführt wurde und die Entfernung zum Objekt sehr viel größer als die Brennweite der Kamera ist.

| Kamerabewegung | Geometrische Bild-Transformation |
|---|---|
| Rotation um optische Achse und/oder Zoom | euklidische Transformation |
| Kleine Kamerabewegung und Objektweite >> Brennweite | euklidische/affine Transformation |
| Rotation um Stativ-Achse; Allgemeine Kamerabewegung bei planarer Szene | affine/projektive Transformation |
| Allgemeine Kamerabewegung | lokale Transformation |

Tabelle 6.1: Zuordnung der Kamerabewegungen zur geometrischen Bild-Transformation

Wenn die Kamerarotation klein ist, kann die projektive Transformation mit einer affinen oder sogar euklidischen Transformation angenähert werden.

**Szenenänderungen**

Ein weiterer großer Bereich möglicher Bildvariationen resultiert aus Szeneänderungen. In diesem Zusammenhang seien beispielsweise Objektbewegungen oder Beleuchtungsänderungen genannt. Wenn eine Szenenänderung stattgefunden hat, ist es nicht ohne weiteres

möglich, zwei Bilder einer Bildfolge pixelgenau aufeinander zu registrieren, da sie sich zu sehr unterscheiden. Dennoch muss das Registrierungsverfahren ausreichend robust sein, um bei Szenenänderungen, die nur einen kleinen Teil der Bildvariationen darstellen, die richtige geometrische Transformation wiederzufinden.

### 6.2.4 Übersicht der Registrierungsverfahren

#### Intensitätsbasierte Bildregistrierung

Ein von Szeliski und Shum vorgestelltes vorgestelltes Verfahren [80, 82] geht von einer projektiven Transformation aus, mit der die Bilder überlagert werden können. Das Verfahren bestimmt direkt die acht Parameter der Transformationsmatrix (Homographie) durch ein nicht-lineares numerisches Verfahren.
Es minimiert die Summe der quadratischen Intensitätsdifferenzen (Sum-Squared-Difference SSD) zwischen dem transformierten Quellbild und dem Zielbild. Hierbei muss sichergestellt sein, dass sich die Größe des betrachteten Bereichs nicht ändert, da in der Vergleichsmetrik kein Bezug dazu vorhanden ist.
Die Pixelkoordinaten im transformierten Quellbild fallen in der Regel nicht auf ganzzahlige Werte. Aus diesem Grund werden die Intensitätswerte durch eine bilineare Interpolation im Quellbild bestimmt.
Das Verfahren arbeitet iterativ und bricht bei Unterschreitung eines Schwellwertes oder nach einer vorgegebenen Anzahl an Iterationsdurchläufen ab.
Die folgende Tabelle fasst das intensitätsbasierte Verfahren nach den Kriterien von Brown zusammen.

| | |
|---|---|
| Merkmalsraum | Intensitätswerte der Pixel und auch Intensitätsgradienten |
| Vergleichsmetrik | Summe der quadratischen Fehler |
| Suchraum | alle zulässigen projektiven Transformationen des Quellbildes |
| Suchstrategie | iteratives numerisches Verfahren; Abbruch<br>durch Unterschreiten eines Schwellwertes oder<br>durch Überschreiten der max. zulässigen Iterationsdurchläufe |

Tabelle 6.2: Komponenten des intensitätsbasierten Verfahrens

Der Vorteil des beschriebenen Verfahrens liegt in der vollständigen Bestimmung der projektiven Transformation (8 freie Parameter), mit der die Bilder registriert werden können. Da dieses Verfahren auf iterativen, numerischen Methoden basiert, wird eine erste Anfangslösung benötigt. Diese kann beispielsweise vom Benutzer eingegeben werden, in dem er die Bilder möglichst genau übereinander legt.

### Merkmalbasierte Bildregistrierung

Im allgemeinen wird bei dem Verfahren der merkmalbasierten Bildregistrierung folgende Vorgehensweise verwendet:

1. Bestimmung der Bildmerkmale durch automatische Verfahren oder manuelle Angabe
2. Suche der Merkmalskorrespondenzen, z.B. Passpunkte
3. Ermittlung der geometrische Transformation, Bildüberlagerung und Farbwertinterpolation.

Die Merkmale müssen nicht notwendigerweise manuell eingegebene Punkte sein. Häufig werden in der Bildverarbeitung Merkmale verwendet, die eine stärkere Aussagekraft als Punkte besitzen. Denkbar sind hier z. B.:

- **Ecken:** Ecken sind dadurch charakterisiert, dass sie in einer lokalen Umgebung die stärkste Krümmung des Gradientenfeldes besitzen. Schnitt- oder Berührungspunkte von Kanten sind ein Beispiel hierfür.
- **Kanten:** Kanten stellen dünne Regionen in den digitalen Bildern dar. Entlang dieser Regionen zeigen die Gradienten in die gleiche Richtung.
- **Konturen:** Konturen sind z.B. Verkettungen von Kanten.

Die untenstehende Tabelle 6.3 fasst die verschiedenen Komponenten des merkmalbasierten Verfahrens nach der Klassifizierung von Brown zusammen.

| | |
|---|---|
| Merkmalsraum | extrahierte Bildmerkmale (Ecken, Linien, Konturverläufe) |
| Vergleichsmetrik | Korrelation der Intensitätswerte und Analyse der Merkmalstruktur |
| Suchraum | alle zulässigen projektiven Transformationen des Quellbildes |
| Suchstrategie | Testen aller Kombinationen<br>von jeweils 4 korrespondierenden Eckpunkten [93]<br>Suchbaum [56]<br>Ransac [46] |

Tabelle 6.3: Komponenten des Merkmalbasierten Verfahrens

### Manuelle Passpunktangabe und numerische Instabilitäten

Eine Bildregistrierung auf Basis von manuell eingegebenen Punkte stellt auf dem ersten Blick eine einfache Lösung dar. Vier oder mehr Punkte werden in den jeweiligen Bilder angeklickt und daraus die Homographie $\boldsymbol{H}$ berechnet, siehe Abbildung 6.3. Anschließend wird das zweite Bild, mit z.B. einem bilinearen Interpolationsverfahren transformiert und mit dem ersten Bild überlagert.

Abbildung 6.3: Bildregistrierung mit manueller Eingabe der Passpunkte

Wenn jedoch der Überlappungsbereich der einzelnen Bilder relativ klein ist und weniger als ein Drittel der Bildbreite beträgt, treten numerische Instabilitäten auf, die zu einer ungenauen Bildregistrierung führen. Dieses Problem kann behoben werden, indem alle Eingabedaten, d.h. in diesem Fall die Bildkoordinaten, in dem Intervall $[-1, 1]$ transformiert werden [46]. Weitere und genauere numerische Analysen, die auch aufwendigere Transformationen der Eingabedaten erfordern, sind z.B. in [52] beschrieben.

#### Analyse im Frequenzraum

Eine klassische Methode stellt die Bildregistrierung auf Basis des Fourier-Ansatzes, auch Phasenkorrelation genannt, dar. Dabei wird die 2D-Verschiebung zweier Bilder bestimmt [16]. Weitere Parameter, wie z.B. die Rotation um die Z-Achse und die Skalierung, können mit diesem Verfahren zurückgewonnen werden [73].
In der untenstehenden Tabelle wird das Verfahren in Bezug zu den Kriterien von Brown gestellt.

| | |
|---|---|
| Merkmalsraum | Fouriertransformierte Bilder |
| Vergleichsmetrik | Kreuzleistungsspektrum |
| Suchraum | euklidische Transformation (Translation, Rotation, Skalierung) |
| Suchstrategie | Direkte Lösung |

Tabelle 6.4: Komponenten des Fourierbasierten Verfahrens

### 6.2.5 Auswahl des Verfahrens

Im Rahmen der Untersuchungen wurden zwei unterschiedliche Registrierungsverfahren ausgewählt und implementiert. Dabei wurden Verfahren, die nicht extrahierte Merkmale

sondern das ganze Bild auswerten, bevorzugt. Durch diese Forderung wird keine Annahme über die Struktur der Szene benötigt und eine höhere Robustheit bezüglich lokaler Veränderungen, wie beispielsweise Schattierungen oder Verdeckungen, erzielt.
Das erste ausgewählte Verfahren arbeitet direkt mit den Intensitätswerten und minimiert die Intensitätsunterschiede der Bilder. Mit Hilfe des Verfahrens wird der spezielle Fall einer Kamera auf einer Stativachse analysiert und eine Echtzeit-Registrierung umgesetzt. Hierfür wird die 2D-projektive Transformation verwendet, siehe Tabelle 6.1.
Das zweite Verfahren beruht auf einer Fouriertransformation der zu registrierenden Bilder. Durch mehrfache Anwendung des Phasenkorrelationsverfahrens ist es hierbei möglich, die euklidische Transformation der beiden Bilder zu bestimmen.
Beide Methode werden in den nächsten Abschnitten im Detail erläutert und evaluiert. Dabei werden die Verfahren in erster Linie hinsichtlich der Kriterien "Zuverlässigkeit" und "Echtzeit-Fähigkeit" analysiert.

## 6.3 Intensitätsbasierte Registrierung

### 6.3.1 Intensitätsunterschiede

Das Ziel der Registrierung ist die automatische Bestimmung der Homographie, siehe Abschnitt 6.3.2. Das Verfahren, das hierbei zum Einsatz kommt, basiert auf der Minimierung der Intensitätsunterschiede beider Bilder. Dabei wird ein Bild solange transformiert bis die Intensitätsunterschiede minimal sind. Als Fehler wird die Quadratsumme der Intensitätsdifferenzen (SSD) gewählt. Die Fehlerfunktion ist folgendermaßen definiert:

$$S_{SSD} = \sum_{x,y} (I_1(x,y) - I_2(\boldsymbol{H}(x,y)))^2 \tag{6.4}$$

Dieses Verfahren weist den Vorteil auf, dass keine Punktkorrespondenzen in den Bildern gefunden werden müssen. Das Auffinden von Punktkorrespondenzen ist für Echtzeitanwendungen oft zu langsam, da schon die Extraktion bekannter Merkmale Zeitprobleme bereitet. Demgegenüber ist die Anzahl der Gleichungen sehr hoch, aber lässt sich, wie in Abschnitt 6.3.3 ausgeführt wird, deutlich reduzieren. Durch die Datenreduktion wird dann die nötige Geschwindigkeit bei der Suche des Minimums erreicht.

### 6.3.2 Verwendete Homographien

Das Minimum wird mit einem Optimierungsverfahren für nichtlineare Gleichungen ermittelt, welches die Parameter der Homographie mit der besten Bildüberlagerung bestimmt. Das genannte Optimierungsverfahren wird im Abschnitt 6.3.5 genauer erläutert.
Eine Homographie wird durch eine $3 \times 3$-Matrix beschrieben und hängt somit von acht Parametern ab. Die Kamerabewegungen, die in diesem Fall betrachtet werden, sind rein rotativ, d.h. sie stammen von einer Kamera auf einem Stativ, wodurch nur Drehungen um die $X$- und $Y$-Achse der Kamera möglich sind.
Für diese Drehungen können die Homographien durch nur zwei Parameter beschrieben werden. Eine korrekte und minimale Parametrisierung ist bei Schätzverfahren, wie beispielsweise der nichtlinearen Optimierung, wesentlich, da diese Parametrisierung zu einer schnelleren Konvergenz führt und lokale Minima vermieden werden können.

Die Homographien der Bewegungen werden wie folgt aufgestellt:

Homographie für Drehung um die $X$-Achse ohne Änderung der Brennweite:

$$\begin{aligned} \boldsymbol{H} &= \boldsymbol{A}\boldsymbol{R}_X\boldsymbol{A}^{-1} = \boldsymbol{A}\begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\omega) & -sin(\omega) \\ 0 & sin(\omega) & cos(\omega) \end{pmatrix}\boldsymbol{A}^{-1} \\ &= \begin{pmatrix} 1 & sin(\omega)\frac{c_x}{fs_y} & -sin(\omega)\frac{c_yc_x}{fs_y} + cos(\omega)c_x - c_x \\ 0 & cos(\omega) + sin(\omega)\frac{c_y}{fs_y} & -sin(\phi)\frac{c_yc_y}{fs_y} - sin(\omega)fs_y \\ 0 & sin(\phi)\frac{1}{f_y} & -sin(\omega)\frac{c_y}{fs_y} + cos(\omega) \end{pmatrix} \end{aligned} \tag{6.5}$$

Homographie für Drehung um $Y$-Achse ohne Änderung der Brennweite:

$$\begin{aligned} \boldsymbol{H} &= \boldsymbol{A}\boldsymbol{R}_y\boldsymbol{A}^{-1} = \boldsymbol{A}\begin{pmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{pmatrix}\boldsymbol{A}^{-1} \\ &= \begin{pmatrix} cos(\phi) - sin(\phi)\frac{c_x}{fs_x} & 0 & sin(\phi)fs_x + sin(\phi)\frac{c_xc_x}{fs_x} \\ -sin(\phi)\frac{c_y}{fs_x} & 1 & sin(\phi)\frac{c_yc_x}{fs_y} + cos(\phi)c_y - c_y \\ -sin(\phi)\frac{1}{fs_x} & 0 & sin(\phi)\frac{c_x}{fs_x} + cos\phi \end{pmatrix} \end{aligned} \tag{6.6}$$

Homographie für Drehung um die $X$- und die $Y$-Achse:

$$\begin{aligned} \boldsymbol{H} &= \boldsymbol{A}\boldsymbol{R}_y\boldsymbol{R}_x\boldsymbol{A}^{-1} = \boldsymbol{A}\begin{pmatrix} cos(\phi) & sin(\omega)sin(\phi) & cos(\omega)sin(\phi) \\ 0 & cos(w) & -sin(\omega) \\ -sin(\phi) & sin(\omega)cos(\phi) & cos(\omega)cos(\phi) \end{pmatrix}\boldsymbol{A}^{-1} \\ &= \begin{pmatrix} cos(\phi) & sin(\omega)sin(\phi)\frac{fs_x}{fs_y} & -cos(\phi)c_x - sin(\omega)sin(\phi)\frac{fs_x}{fs_y} + cos(\omega)cos(\phi)fs_x \\ -sin(\phi)\frac{c_x}{fs_x} & +sin(\omega)cos(\phi)\frac{c_x}{fs_y} & +sin(\phi)\frac{c_xc_x}{fs_x} - sin(\omega)cos(\phi)\frac{c_xc_y}{fs_y} + cos(\omega)cos(\phi)c_x \\ -sin/\phi) * \frac{c_y}{fs_x} & cos(\omega) & -cos(\omega)c_y - sin(\omega)fs_y + sin(\phi)\frac{c_xc_x}{fs_x} \\ & +sin(\omega)cos(\phi)\frac{c_y}{fs_y} & -sin(\omega)cos(\phi)\frac{c_xc_y}{fs_y} + cos(\omega)cos(\phi)c_y \\ -sin(\phi)\frac{1}{fs_x} & sin(\omega)cos(\phi)\frac{1}{fs_y} & sin(\phi)\frac{c_x}{fs_x} - sin(\omega)cos(\phi)\frac{c_y}{fs_y} + cos(\omega)cos(\phi) \end{pmatrix} \end{aligned} \tag{6.7}$$

Homographie ohne Drehung mit einer Änderung der Brennweite:

$$\boldsymbol{H} = \boldsymbol{A}\boldsymbol{A'}^{-1} = \begin{pmatrix} \frac{f}{f'} & 0 & c_x - \frac{f}{f'}c_x \\ 0 & \frac{f}{f'} & c_y - \frac{f}{f'}c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{6.8}$$

Die Benutzung der realen Parameter bei der Aufstellung der Homographie hat nicht nur den Vorteil der Zeitersparnis, sondern ermöglicht auch die schnelle und einfache Erfassung der Bewegungsparameter.

### 6.3.3 Datenreduktion

Bei der Orientierung der Bilder mit Hilfe der Intensitätsdifferenzen entsteht eine sehr große Anzahl an Gleichungen, für die das Intensitätsminimum gesucht werden muss. Dabei entsteht für jedes Pixel und für jeden Farbwert eine Gleichung.

$$Anzahl\ der\ Gleichungen = Pixelzahl_{Zeilen} * Pixelzahl_{Spalten} * Anzahl\ der\ Farben$$

Da die Dauer der Suche des Minimums der Farbdifferenzen über die Bewegungsparameter stark von der Anzahl der Gleichungen abhängt, stellt sich die Frage, wie man die Zahl der Gleichungen ohne einen Genauigkeitslust verringern kann.
Für die Datenreduktion können die folgenden Ansätze betrachtet werden:

- **Teilbereiche:** Es wird nur ein Teil des Bildes zur Berechnung herangezogen. Damit verringert sich die Zahl der Zeilen bzw. Spalten. Diese Methode wurde z.B. in [51] eingesetzt, wobei die Zahl der Zeilen versuchsweise sogar bis auf eine reduziert werden konnte.

- **Grauwertbild:** Ein Farbbild wird in ein Grauwertbild umgewandelt. Dabei werden alle Farbauszüge zu einem Wert zusammengefasst. Dies reduziert die Zahl der Gleichungen um die Anzahl der Farben.

- **Zoom:** Das Bild wird um einen Faktor $> 1$ verkleinert. Es wird zum Beispiel jede zweite Zeile und jede zweite Spalte entfernt. Zur Steigerung der Genauigkeit kann auch so vorgegangen werden, dass alle Werte zu einem Wert gemittelt und so alle Werte einbezogen werden.

- **Streifen:** Ein anderer Ansatz zur Datenreduktion ist das Zusammenfassen von Daten in Streifen. Dabei werden die Farbwerte in Form von Streifen, die senkrecht zum optischen Fluss angeordnet werden, zu einem Wert komprimiert. Die Berechnung solcher Streifen wird in [77] exakt definiert. Bei einer Verschiebung in $X$-Richtung werden die Daten beispielsweise spaltenweise aufsummiert.

- **Bildpyramiden:** Der Bildpyramiden-Ansatz geht von einer stufenweisen Verfeinerung der Lösung aus. Zuerst wird mit sehr groben Bildern eine gute erste Näherungslösung gesucht. Daraufhin wird die Auflösung gesteigert, bis die gewünschte Genauigkeit erreicht ist.

Hinsichtlich der Verfahren und ihrer Kombination stellt sich die Frage, wie stark die Genauigkeit bei solchen Vereinfachungen abnimmt. Dabei ist zu unterscheiden, ob nur die Anzahl der Daten oder ob die maximal erreichbare Genauigkeit abnimmt. Bei jeder Datenreduktion gehen Informationen verloren, und die Bestimmung des realen Minimums wird schwieriger. Wird die Auflösung eines Bildes reduziert oder ein Bild um einen bestimmten Zoomfaktor verkleinert, sinkt auch die erreichbare Genauigkeit.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Muster: | 10 | 3 | Bild: | 10 | 15 | 12 | 10 | 4 | 13 | 2 | 10 | 15 | 3 |
| | 12 | 8 | | 12 | 7 | 10 | 11 | 7 | 10 | 8 | 12 | 3 | 8 |
| | 10 | 15 | | 1 | 4 | 10 | 10 | 15 | 10 | 15 | 10 | 8 | 15 |
| $\sum$: | 32 | 26 | $\sum$: | 31 | 26 | 32 | 31 | 26 | 33 | 25 | 32 | 26 | 26 |

Das Ziel ist, die Position zu finden, an der das Suchmuster am besten in den Suchbereich passt. Dafür wird das Suchmuster von links nach rechts über den Suchbereich verschoben und das Minimum der Summe der Quadratdifferenzen ermittelt. Im ersten Schritt werden alle Werte berücksichtigt, dann nur die Spaltensummen und zum Schluss wird jede Zeile einzeln betrachtet.

| Spaltenposition | SSD aller Werte | SSD der Spaltensumme | SSD Zeile 1 | SSD Zeile 2 | SSD Zeile 3 |
|---|---|---|---|---|---|
| 1 | 347 | 1 | 144 | 1 | 202 |
| 2 | 196 | 72 | 106 | 29 | 61 |
| 3 | 91 | 25 | 53 | 13 | 25 |
| 4 | 3 | 1 | 1 | 2 | 0 |
| 5 | 205 | 85 | 136 | 29 | 50 |
| 6 | 14 | 2 | 10 | 4 | 0 |
| 7 | 195 | 85 | 113 | 32 | 50 |
| 8 | 218 | 0 | 144 | 25 | 49 |
| 9 | 110 | 36 | 25 | 81 | 4 |

Tabelle 6.5: Verschiedene Methoden der Datenreduktion

In der Tabelle 6.5 werden unterschiedliche Methoden der Datenreduktion dargestellt. Aus dem ersten Bild wird nur ein kleiner Bereich, nämlich das Muster, ausgesucht. Dieses soll dann in das Bild mittels einer Verschiebung wiedergefunden werden. Im ersten Schritt werden alle sechs Werte berücksichtigt, dann nur die Spaltensummen und zum Schluss wird jede Zeile einzeln betrachtet.

| Bildregistrierungsverfahren | Anzahl der Gleichungen (Bilddim.: 600x600) |
|---|---|
| ohne Datenreduktion | 1080000 |
| Grauwertbild für den mittleren Bereich | 9000 |
| Bild um Faktor 10 verkleinert | 900 |
| Spaltensummen | 300 |
| Spaltensummen und Verkleinerung um Faktor 10 | 30 |

Tabelle 6.6: Anzahl der Gleichungen für verschieden Datenreduktionstechniken

Anhand Tabelle 6.6 wird verdeutlicht, dass die vierte Position die beste Bildüberlagerung liefert. Außerdem kann festgestellt werden, dass bei der Datenreduktion auch lokale Minima entstehen, die zur falschen Bildüberlagerung führen. Die Verfahren liefern folglich nur korrekte Lösungen, wenn bei der Überlagerungsauswertung lokale Minima als Fehlerquelle

ausgeschlossen werden.
Hinsichtlich der Genauigkeit wird bei allen Verfahren die Auflösung nicht reduziert, da die Suchrichtung der $X$-Richtung entspricht. Bei einer Verkleinerung des Bildes um einen bestimmten Zoomfaktor sinkt die Auflösung auch um denselben Faktor in $X$-Richtung.
Im Rahmen dieser Arbeit wurde immer nur der mittlere Bereich des Bildes verwendet, da in diesem Fall der Bildausschnitt sicher in beiden Bildern vorhanden ist. Durch diese Auswahl wurde die Anzahl der Gleichungen um den Faktor vier reduziert. Da die Reduzierung auf ein Grauwertbild keinen Auflösungsverlust nach sich zieht, wurden außerdem in der Regel Grauwertbilder verwendet, wodurch die Anzahl der Gleichungen um den Faktor 3 vermindert wird.
Basierend auf den voranstehenden Bildern wurden einfache Skalierungen und die Aufsummierung in Spalten vorgenommen. Bei den Skalierungen wurde nicht interpoliert sondern Bildzeilen und -spalten gelöscht. Als Streifen wurden Zeilen- bzw. Spaltensummen gebildet, mit denen Bildverschiebungen in $X$- bzw. $Y$-Richtung und dadurch auch die 3D-Kamerarotationen um die $X$- und $Y$-Achsen ermittelt wurden. Bei der Ermittlung der Rotationen mit Hilfe dieser Datenreduktion ist zu berücksichtigen, daß die Spalten nicht mehr senkrecht auf dem optischen Fluss stehen und durch diese Abweichung kleine Fehler erzeugt werden. Die Annäherung ist jedoch dadurch gerechtfertigt, daß die Kameradrehung zwischen zwei Bildaufnahmen sehr klein ist und daß das beschriebene Verfahren den Berechnungsgang vereinfacht.
In der Tabelle 6.6 wird für unterschiedliche Methoden die Gleichungsanzahl angegeben. Dabei wird ein Bild mit $600 * 600$ Pixels und drei Farbwerten pro Pixel als Grundlage zugrunde gelegt.

### 6.3.4 Interpolation von Zwischenwerten

Wenn ein Punkt $\boldsymbol{p}$ eines Quellbildes durch eine Homographie in ein Zielbild transformiert wird, ergeben sich in der Regel keine ganzzahligen Pixelwerte für den Punkt $\boldsymbol{p'}$. Da die Bilder mit hohen Skalierungsfaktoren verkleinert sind, wird üblicherweise zur Berechnung der Pixelwerte eine lineare Interpolation verwendet. Auf Grund der groben Rasterung führt dieses Verfahren jedoch, wie in Tabelle 6.7 zu sehen ist, bei der Bildung der SSD nicht zu korrekten Ergebnissen.
Werden die Daten der Grauwerte in dieser Form reduziert, besteht das Problem dass die Quadratsumme durch die lineare Interpolation zwischen zwei Werten kleiner wird und dadurch künstlich lokale Minima verursacht werden, siehe Beispiel Tabelle 6.7.
Die Abbildung 6.4 veranschaulicht noch mal dieses Problem. Die gefundenen Drehwinkel pendeln bei dieser Methode immer zwischen zwei Werten und die graue Kurve der berechneten Winkelwerte verfolgt nur grob und mit viele Aussetzer die schwarzen Kurve der Original-Winkelwerte.

Um das Problem zu lösen, dürfen die Differenzen der Grauwerte nicht nur an einzelnen Punkten bestimmt werden, sondern sie müssen kontinuierlich ermittelt werden. Es muss also eine Integration über alle Positionen erfolgen. Das bedeutet, dass nicht einzelne Grauwertdifferenzen minimiert werden, sondern die Fläche zwischen den Kurven, wie in Abbildung 6.5 zu sehen ist. In der Abbildung ist die zu minimierende Fläche grau eingezeichnet. Zusätzlich sind auch die Positionen mit schwarzen Balken eingezeichnet, an denen vorher die SSD mit den einfachen Grauwertdifferenzen berechnet wurden.

Beispielgrauwerte:

| Spalte | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Grauwerte (a) | 0 | 0 | 10 | 0 | 0 |
| Grauwerte (b) | 0 | 0 | 0 | 0 | 0 |

Wenn die Grauwerte (a) durch die Grauwerte (b) abgebildet werden, resultiert daraus als Intensitätsdifferenz immer 10 und damit für die SSD, da auch alle Zwischenwerte zu 0 interpoliert werden, 100. Wenn jedoch die Grauwerte (b) durch die Grauwerte (a) abgebildet werden, ändert sich das. Solange die Abbildung auf ganzzahlige Spalten erfolgt, ist die Differenz immer noch 10 und die SSD beträgt 100. Wenn aber die Spalte 2 auf 2.5 und die Spalte 3 auf 3.5 abgebildet werden, so werden die fehlenden Grauwerte zu 5 $((10+0)/2)$ interpoliert. Die Grauwertdifferenz ist dann $5+5$ und damit auch wieder 10, die SSD besitzt jedoch eine Intensitätsdifferenz mit dem Wert $5^2+5^2=50$.

Tabelle 6.7: Fehler der SSD bei groben Rastern und linearer Interpolation der Grauwerte (Beispiel)

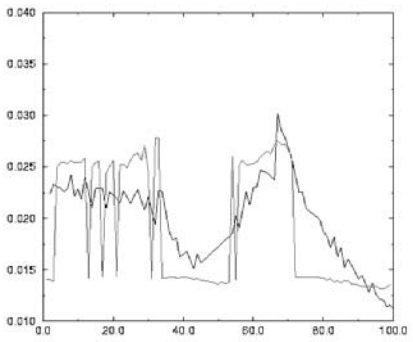



Abbildung 6.4: Zu geringe Genauigkeit bei der linearen Interpolation der SDD

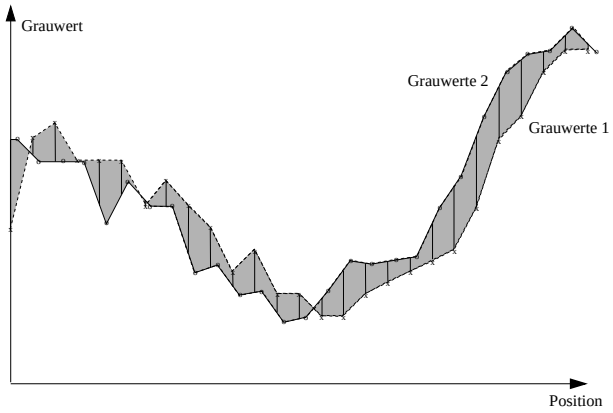



Abbildung 6.5: Berechnung der SSD als Fläche

Die Fläche zwischen den Kurven wird nicht durch ein aufwendiges Integrationsverfahren, sondern durch eine einfache Zerlegung in Dreiecksflächen berechnet. An jeder ganzzahligen Position wird die Fläche unterteilt und getrennt berechnet. Dabei sind, wie in Abbildung 6.6 dargestellt wird, zwei Fälle zu unterscheiden.

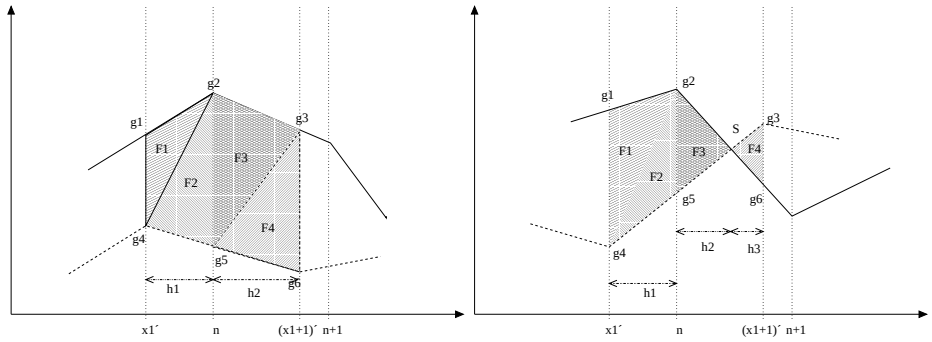



Abbildung 6.6: Berechnung der Fläche zwischen den Grauwertkurven

Im ersten Fall schneiden sich die Grauwertkurven, im zweiten Fall liegen zwischen den kurven kein Schnittpunkt vor, siehe Abbildung 6.6. Die einzelnen Flächen sind Dreiecke und können leicht mit

$$F = \frac{1}{2}(g_n - g_m) * h$$

berechnet werden. Wenn sich die Kurven schneiden, muss der Schnittpunkt $\boldsymbol{S}$ und $h2$ und $h3$ berechnet werden.

### 6.3.5 Minimierungsverfahren

Das Ziel des Minimierungsverfahren ist, die Parameter der Homographie zu bestimmen, die die Fläche zwischen den Grauwertkurven minimieren. Dabei wird davon ausgegangen, daß die kleinstmögliche Flächenabweichung die beste Bildüberlagerung liefert. Die Lösung wird mit Hilfe eines Gradienten-Verfahrens bestimmt, wobei die zulösende Gleichung folgende Form besitzt:

$$\epsilon = min \sum_i e_i^2 \tag{6.9}$$

wobei $e_i$ der Grauwertdifferenz an der Position $i$ entspricht.
Eine Standardmethode für nichtlineare Gleichungen bietet der Levenberg-Marquard-Algorithmus [69, 61, 55]. In dieser Arbeit wird die Implementierung der Fortran Librarie MINPACK[3] verwendet.

### 6.3.6 Vergleich zwischen “Zoom”- versus “Streifen”-Ansatz

Im Rahmen dieser Arbeit werden zwei unterschiedliche Ansätze “Zoom” und “Streifen” zur Datenreduktion untersucht. Die erste “Zoom”-Technik beruht auf einer einfachen Skalierung der Bilder. Die zweite Lösung bildet Streifen, die durch die Aufsummierung der Intensitätspixelwerte entstehen.

[3] http://www.netlib.org

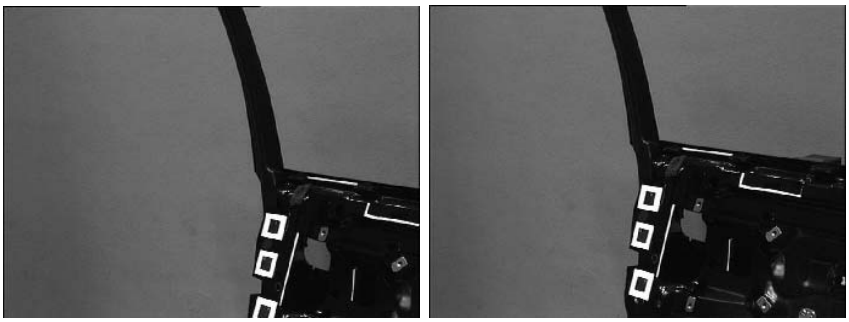


Abbildung 6.7: Testbilder aus dem AR-Szenarios “Tür-Montage”

Für beide Ansätze wurden Ausschnitte aus dem mittleren Bereich der Bilder verwendet. Dabei wurde an allen Rändern jeweils 25% der Bildhöhe bzw. Bildbreite abgeschnitten und die Bilder in Grauwertbilder umgewandelt.
Die Datenreduktion mit Hilfe einer einfache Skalierung der Bilder besitzt den Vorteil, dass das Verfahren allgemein einsetzbar bleibt und alle Homographie-Typen erfasst werden können. Allerdings ist die Anzahl der Gleichungen für eine Minimierung sehr hoch. Dagegen ermöglicht der Streifen-Ansatz eine erhebliche Datenreduktion, wobei jedoch nur eine Bewegungsrichtung bestimmt werden kann. Dies bedeutet, dass für zwei Drehungen zwei getrennte Minimierungen durchgeführt werden müssen.

Abbildung 6.8: Aus den Testbildern 6.7 erzeugtes Mosaik

Der Versuch, die Daten bei der Skalierung nicht zusammenzufassen, sondern einfach nur jedes $n$-te Pixel zu verwenden, brachte keinen Erfolg, da dabei kleine Fehler, wie zum Beispiel Schatten, große Auswirkungen haben. Außerdem fällt dabei der Vorteil der Tiefpassfilterung, der in Abschnitt 6.3.9 erläutert wurde, aus.
Die Abbildungen 6.9 und 6.10 zeigen die skalierten Grauwertbilder bzw. -kurven und die dazugehörigen SSD. Ein eindeutiges Minimum kann bei beiden Kurven erkannt werden.

Anhand der Ausgangsbilder, Abbildung 6.7, kann hinsichtlich der Eingabebilder deutlich ein Qualitätsverlust erkannt werden.

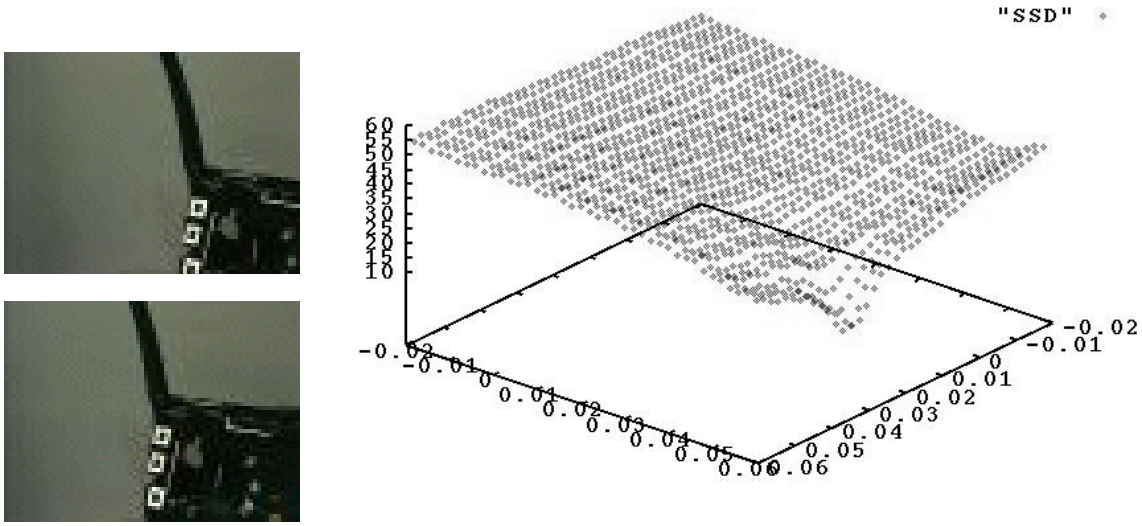



Abbildung 6.9: SSD für eine einfache Skalierung und eine Drehung um die $X$- und $Y$-Achse

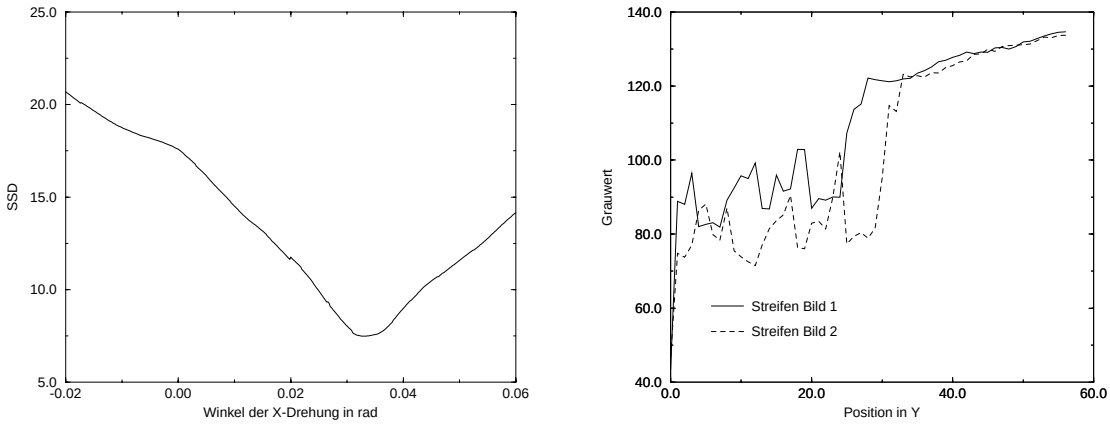



Abbildung 6.10: SSD für den Winkel $\omega$ und die dazugehörigen Streifenwerte

In der Tabelle 6.8 sind die erzielten Ergebnisse für die Winkel $\omega$ und $\phi$ dargestellt. Dabei wurden unterschiedliche Skalierungsfaktoren zugrunde gelegt. Wenn die Minimierung gescheitert ist, wurde in der Tabelle kein Winkel eingetragen.

| Skalierungsfaktor | Winkel in Grad für Streifen | Winkel in Grad für eine einfache Skalierung |
|---|---|---|
| 1 | $\omega = -1.86\ \phi = 3.24$ | $\omega = -\ \phi = -$ |
| 5 | $\omega = -1.86\ \phi = 3.27$ | $\omega = -\ \phi = -$ |
| 8 | $\omega = -1.90\ \phi = 3.23$ | $\omega = -\ \phi = -$ |
| 10 | $\omega = -1.90\ \phi = 3.24$ | $\omega = -1.73\ \phi = 2.62$ |
| 15 | $\omega = -2.05\ \phi = 3.24$ | $\omega = -1.60\ \phi = 3.14$ |
| 20 | $\omega = -2.13\ \phi = 3.14$ | $\omega = -2.09\text{d}\ \phi = 4.35$ |

Tabelle 6.8: Berechnete Winkel für die Testbilder "Autotür"

Wie zu erwarten ist, nimmt einerseits die Genauigkeit mit steigenden Skalierungsfaktoren ab und andererseits liefert die Skalierungsmethode, aufgrund der lokalen Minima, erst bei höheren Faktoren Ergebnisse.

### 6.3.7 Genauigkeitssteigerung durch ein hierarchisches Verfahren

Um die Genauigkeit zu steigern und vor allem die Problematik der lokalen Minima zu verringern, werden hierarchische Verfahren eingesetzt. Dabei wird mit einer geringen Auflösung begonnen, um erste Näherungswerte für die Lösung zu erhalten. Mit dieser Näherungslösung wird dann die Suche bei einer höheren Auflösung fortgesetzt.
Dieses Verfahren ist für das Testbild "Autotür" aus Abbildung 6.7 mit einer einfachen Skalierung getestet worden. Wenn keine Skalierung durchgeführt wird, wird die maximale Auflösung und damit die höchste Genauigkeit erzielt.
In Tabelle 6.9 ist das Verfahren für Skalierungsfaktoren von 1 bis 40 dargestellt. Die errechneten Winkel werden als Startwerte für die Minimierung bei der niedrigeren Skalierungsstufe angesetzt.

| Skalierungsfaktor | Winkel in Grad |
|---|---|
| 40 | $\omega = -0.11\ \phi = 2.7$ |
| 30 | $\omega = -1.7\ \phi = 3.2$ |
| 20 | $\omega = -2.0\ \phi = 3.5$ |
| 10 | $\omega = -1.8\ \phi = 3.2$ |
| 1 | $\omega = -1.8\ \phi = 3,3$ |

Tabelle 6.9: Mit hierarchischen Verfahren berechnete Winkel

Die Tabelle 6.9 zeigt, dass die Winkel gegen die endgültige Lösung konvergieren und die Genauigkeit mit jedem Näherungsschritt zunimmt. Zu große Schritte zwischen den Skalierungsfaktoren können jedoch nicht eingesetzt werden. Ein direkter Übergang vom Skalierungsfaktor 40 zum Skalierungsfaktor 20 führt beispielsweise für $\omega$ zu keiner Lösung.

In Tabelle 6.10 sind die für unterschiedliche Startwerte berechneten Winkel für $\omega$ bei einem Skalierungsfaktor von 20 angegeben.

| Startwinkel in Grad | errechneter Winkel in Grad |
|---|---|
| -0.57 | $\omega = -1.14$ |
| -1.15 | $\omega = -1.20$ |
| -1.72 | $\omega = -2.01$ |
| -2.29 | $\omega = -2.01$ |
| -2.87 | $\omega = -2.92$ |

Tabelle 6.10: Für unterschiedliche Startwerte berechnete Winkel

Der Startwert muss sich bei einem Skalierungsfaktor von 20 sehr nah an der richtigen Lösung befinden, damit korrekte Ergebnisse erzielt werden.
Solche hierarchischen Verfahren sind sehr genau, aber sie werden, da mehrere aufwendige Minimierungen durchgeführt werden müssen, vor allem für Off-line Anwendungen angewendet. Des weiteren kann festgestellt werden, dass lokale Minima kein Schwierigkeit bereiten, da die Winkel zwischen zwei Bildern sehr klein sind und die Initiallösung schon sehr nah an der richtigen Lösung liegt.

### 6.3.8 Kamera-Tracking auf einem Stativ

In diesem Abschnitt wird untersucht, ob es mit den intensitätsbasierten Bildregistrierungsmethoden möglich ist, die Bewegungen einer Kamera ohne die Stützung von Markern mit einer ausreichenden Genauigkeit zu berechnen. Dabei werden, entsprechend den Freiheitsgraden einer Kamera auf einem Stativ, die Drehungen um die $X$- und die $Y$-Achse ermittelt.

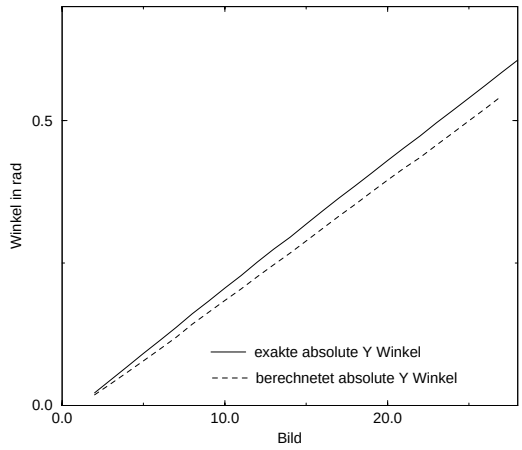



Abbildung 6.11: Drift zwischen den berechneten und tatsächlichen Winkeln

Wie aus der Untersuchung hervorgeht, ist das Verfahren in seiner bisherigen Form noch nicht einsetzbar. Die Fehler zwischen den einzelnen Bildern sind zwar klein, aber für die Erweiterung der Szene sind die absoluten Parameterwerte entscheidend. Aufgrund der sequenziellen Vorgehensweise (Tracking von Bild zu Bild) summieren sich die Fehler auf und verursachen dadurch ein sogenannter Drift der zu berechneten Parameter. In Abbildung 6.11 wird dieses Phänomene veranschaulicht. Die berechneten absoluten Winkel (durchgezogene Linie) entfernen sich über die Zeit kontinuierlich von den richtigen Winkeln (geschtrichelte Kurve).
Um dieses Problem zu umgehen, muss entweder zur Stabilisierung wieder auf Marken zurückgegriffen oder ein Referenzbild (oder Referenzmosaik) verwendet werden. Bei der Verwendung eines Referenzbildes wird der Winkel nicht mehr zwischen zwei aufeinanderfolgenden Bildern berechnet, sondern der Winkel zwischen dem Referenzbild und dem aktuellen Live-Videobild, siehe Abschnitt 6.1.2. Damit wird der absolute Winkel in Bezug auf das Referenzbild immer wieder neu berechnet, was die Aufsummierung der Fehler und das Drift-Problem verhindert. Dieses Verfahren wird im Abschnitt 6.4 beschrieben.

### 6.3.9 Anforderungen an die Bilder

Das Konvergieren des Minimierungsverfahrens hängt stark vom Bildinhalt ab. Das Verfahren wird beispielsweise sehr anfällig, wenn die Bilder viele scharfen Kanten enthalten. Im Idealfall sollte die Intensitätsdifferenz mit dem Abstand der Bilder kontinuierlich steigen. Üblicherweise ist die Intensitätsdifferenz jedoch sehr starken Schwankungen unterworfen und die SSD weist deshalb kein eindeutiges Minimum auf. In Bildern mit starken Intensitätsschwankungen sind viele hohe Frequenzen vorhanden, die eine hohe Anzahl an lokalen Minima verursachen und das Verfahren stören. In Abbildung 6.13 sind die Grauwertkurven und die SSD für die Testbilder aus Abbildung 6.12, die starke Intensitätsschwankungen aufweisen, aufgezeichnet.

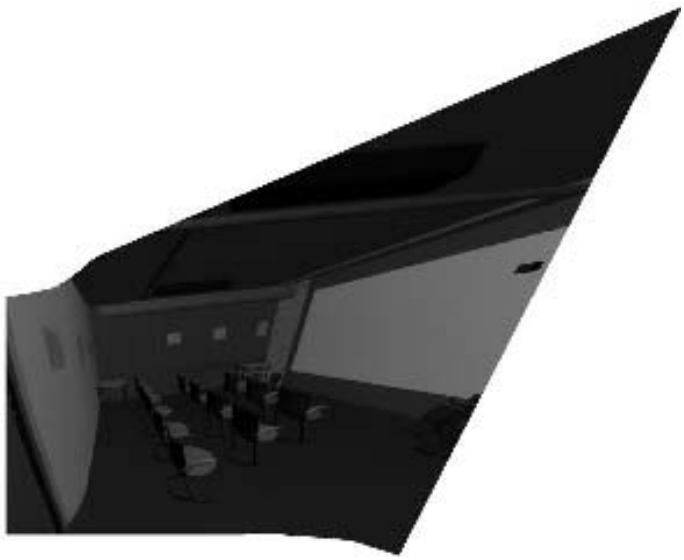
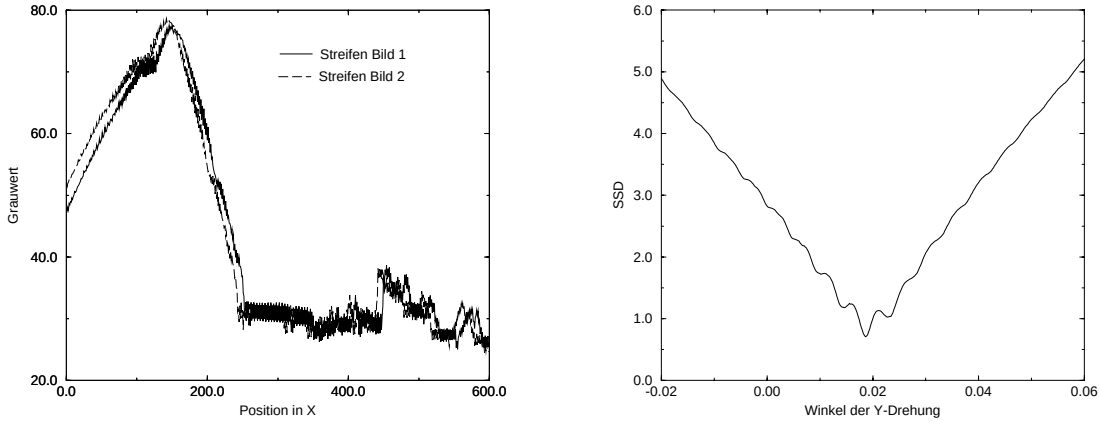


Abbildung 6.12: Aus bekannten Kalibrierungsdaten erstelltes Referenzmosaik




(a) (b)

Abbildung 6.13: SSD bei hohen Frequenzen; (a) die dazugehörigen Streifen und (b) die SSD

An dieser Stelle macht sich ein weiterer Vorteil der Skalierung der Bilder bemerkbar. Durch das Zusammenfassen von Pixels wird gleichzeitig eine Tiefpassfilterung durchgeführt und hohe Frequenzen werden somit ausgeschlossen. Die Konvergenz des Minimierungsverfahrens ist dadurch verbessert, und Startwerte, die weiter von der richtigen Lösung liegen, können zugelassen werden.
In Abbildung 6.14 ist der mögliche Konvergenzbereich dargestellt. Wenn der Startwert außerhalb des Konvergenzbereichs liegt, wird ein falsches Ergebnis errechnet.

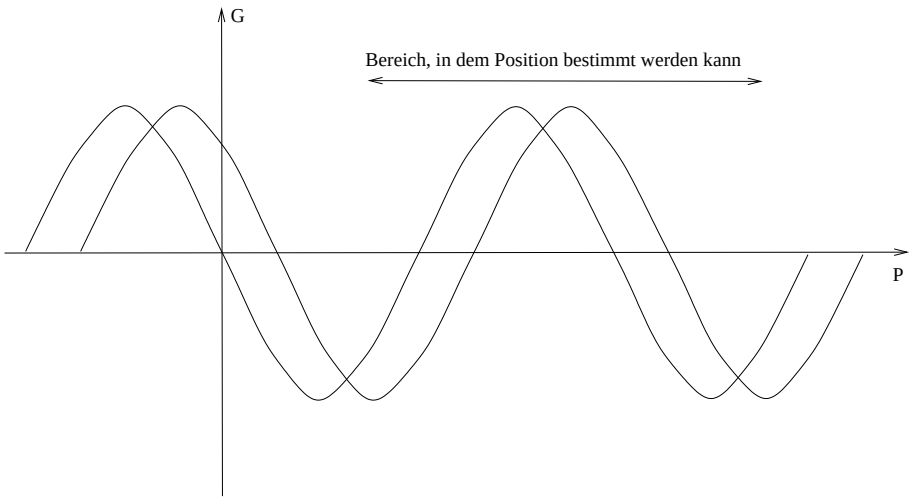
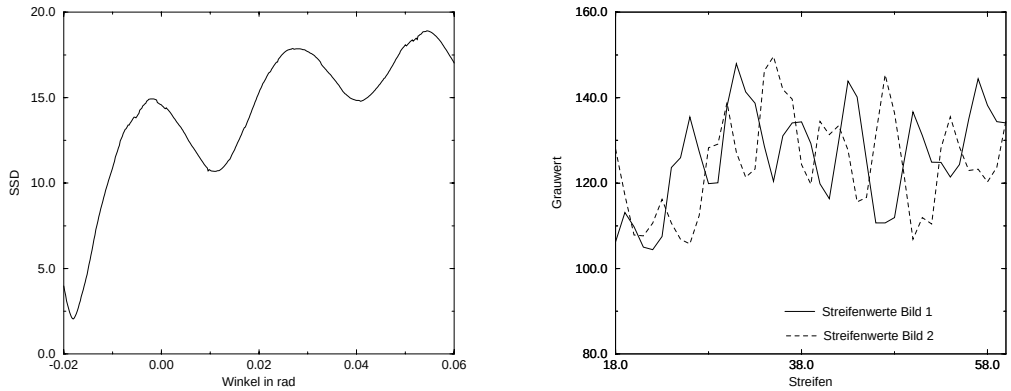



Abbildung 6.14: Konvergenzbereich bei Minimierung

Der Konvergenzbereich wird von den markanten Strukturen (Regionen) im Bild bestimmt. Für die Autotür aus Abbildung 6.7 sind zwei dominante Regionen, die Tür und der Hinter-

grund, entscheidend. Die Kurve der Intensitätsdifferenz steigt kontinuierlich, je größer der Abstand zwischen den beiden Bildern wird. Der Konvergenzbereich ist eindeutig definiert und die Kameraorientierung kann zuverlässig berechnet werden.
In der Abbildung 6.15 wiederholt sich das Hauptbildmuster und erzeugt mehrere lokale Minima. Der Suchbereich , der durch die Bäume vorgegeben ist, muss in diesem Fall stark eingeschränkt werden. Wenn der Startwinkel von der richtigen Lösung zu stark abweicht, bricht die Minimierung in einem lokalen Minimum, das der Deckung der Bäume entspricht, ab. Im Bild 6.15 überlagern sich die Bäume optisch und der Rest des Bildes kommt nicht zur Deckung. Dieses Problem ist an der SSD in Abbildung 6.16 deutlich zu erkennen. Bei einem Startwinkel von 0 Grad wird das lokale Minimum (bei 0.01) und nicht das absolute Minimum (bei $-0.018$) gefunden.

Abbildung 6.15: Falschregistrierung verursacht durch ein lokales Minimum




(a) (b)

Abbildung 6.16: (a) SSD) und (b) Streifenwerte für die Abbildung 6.15

### 6.3.10 Untersuchungen des gewähltes intensitätsbasiertesn Registrierungsverfahren

In diesem Abschnitt werden die erzielten Ergebnisse dargestellt und bewertet. Dabei werden verschiedene Methoden zur Datenreduktion verglichen und günstige Werte für die Skalierung und die Parameter der Minimierung ermittelt.
Um die Funktionen und Parameter zu testen, wird unterschiedliches Bildmaterial eingesetzt. Zum einen werden künstliche Bilder verwendet, um die Genauigkeit der Ergebnisse ermitteln zu können, zum anderen werden reale Bilder benutzt, um die spätere Anwendbarkeit in verschiedenen Umgebungen sicherzustellen. Künstliche Bilder besitzen den Vorteil, dass die interne, sowie die externe Kamerakalibrierung exakt bekannt ist und daher die erzielten Ergebnisse überprüft werden können. Für echte Bilder erfolgt eine visuelle Überprüfung anhand der erstellten Bildmosaike. Diese Methode ist jedoch, wie sich anhand der Untersuchungen herausgestellt hat, nicht sehr genau, da Fehler von $1-2$ Pixels zwischen den Bildern für den Menschen kaum wahrnehmbar sind.
Für die Erstellung von Bildmosaiken wurde zunächst eine auf einem Stativ montierte Kamera eingesetzt, wodurch die Kamerabewegungen auf Drehungen um die $X$- und $Y$-Achse beschränkt werden.

#### Datenreduktion

In diesem Abschnitt wird die intensitätsbasierte Bildregistrierung im Hinblick auf unterschiedliche Techniken der Datenreduktion untersucht. Die erreichbare Winkelgenauigkeit und die Qualität der Minimierung wird anhand der SSD-Kurven analysiert. Wenn diese Kurven ein eindeutiges Minimum, das bei einer größeren Skalierung nicht verloren geht, aufweisen, liefert die Minimierung zu korrekten Ergebnissen. In Abbildung 6.17 sind die SSDs für unterschiedliche Skalierungsfaktoren bei der Verwendung des Streifenansatzes dargestellt. Die Position des Minimums ändert sich kaum bei einer Variation des Skalierungsfaktors, und das Minimum bleibt eindeutig bestimmt. Erst bei einer Skalierung mit dem Faktor 15 verschiebt es sich merklich und ein Genauigkeitsverlust wird merkbar. Des weiteren ist zu erkennen, dass die Kurven mit einem höheren Skalierungsfaktor glatter werden, wodurch die Konvergenz des Minimierungsverfahrens unterstützt wird.

Für die gefundenen kritischen Reduktionsfaktoren 10 und 15 wird anschließend die Streifen- und die Skalierungsmethode miteiander verglichen. Die Ergebnisse beider Methoden sind in Tabelle 6.11 für verschiedene Testbilder ("Lab", "Door" und "Expo") dargestellt.
Wie der Tabelle zu entnehmen ist, lifert die Datenreduktion durch Skalierung ungenauere Ergebnisse. Außerdem müssen exaktere Startwerte und höhere Skalierungsfaktoren gewählt werden, um die Konvergenz des Minimierungsverfahren zu gewähren, wodurch die Genauigkeit wiederum sinkt.

#### Genauigkeit

Um die erreichte Genauigkeit zu untersuchen, wurden künstliche Bilder angewendet. Abbildung 6.12 zeigt das Mosaik, das mit Hilfe der bekannten Kalibrierungsdaten erstellte wurde. In Abbildung 6.18 ist ein automatisch erstelltes Mosaik der gleichen Bildsequenz zu sehen, wobei der Streifenansatz zur Datenreduktion verwendet wurden. Es wurden nur

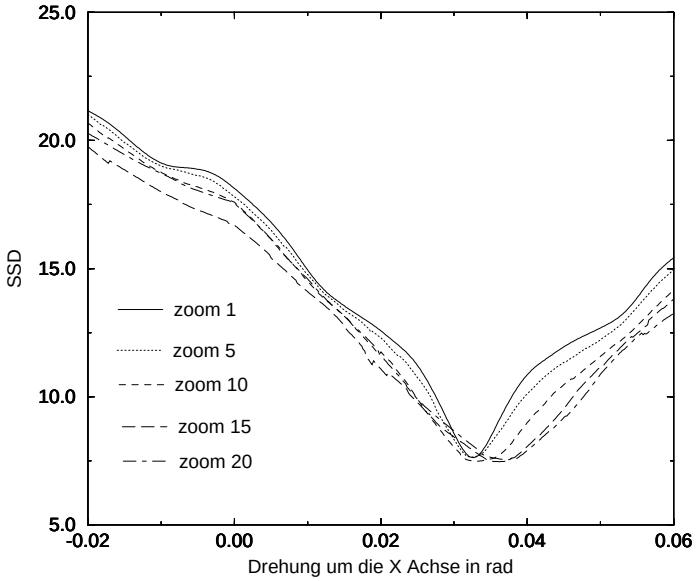



Abbildung 6.17: SSD für die $X$-Drehung mit unterschiedlichen Skalierungsfaktoren

| Bild | Größe | Skalierungs-faktor | Datenreduktion | Winkel in Grad |
|---|---|---|---|---|
| Lab | 600x600 | 10 | Streifen | $\omega = -1.55\ \phi = 1.08$ |
| Lab | 600x600 | 10 | Zoom | $\omega = -1.22\ \phi = 1.15$ |
| Lab | 600x600 | 15 | Streifen | $\omega = -0.46\ \phi = 1.10$ |
| Lab | 600x600 | 15 | Zoom | $\omega = -1.28\ \phi = 0.88$ |
| Door | 768x576 | 5 | Streifen | $\omega = -1.86\ \phi = 3.24$ |
| Door | 768x576 | 5 | Zoom | $\omega = -\ \phi = -$ |
| Door | 768x576 | 15 | Streifen | $\omega = -2.05\ \phi = 3.24$ |
| Door | 768x576 | 15 | Zoom | $\omega = -1.55\ \phi = 3.11$ |
| Expo | 720x288 | 5 | Streifen | $\omega = 0\ \phi = 1.51$ |
| Expo | 720x288 | 5 | Zoom | $\omega = 0\ \phi = 1.52$ |
| Expo | 720x288 | 10 | Streifen | $\omega = -0.06\ \phi = 1.46$ |
| Expo | 720x288 | 10 | Zoom | $\omega = 0\ \phi = 1.32$ |
| Expo | 720x288 | 15 | Streifen | $\omega = -0.08\ \phi = 1.49$ |
| Expo | 720x288 | 15 | Zoom | $\omega = -0.001\ \phi = 0.98$ |

Tabelle 6.11: Berechnete Winkel für unterschiedliche Techniken der Datenreduktion

die ersten 27 von 100 Bilder benutzt, weil das Mosaik ansonst durch die perspektivische Verzerrung zu groß geworden wäre.

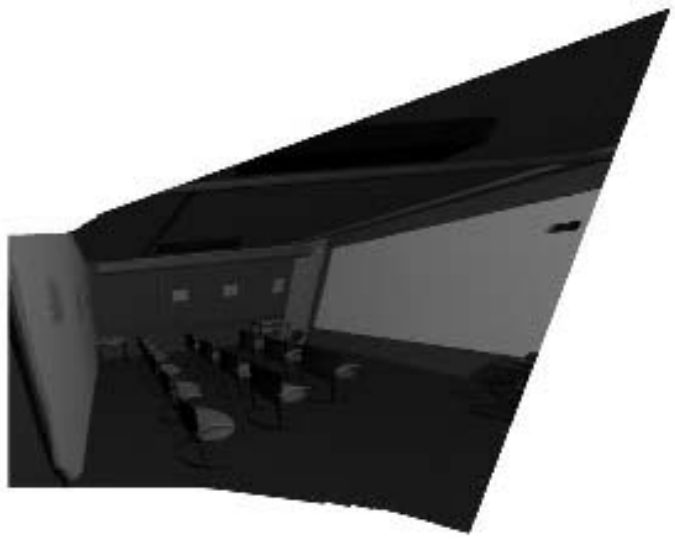


Abbildung 6.18: Aus 27 Bildern automatisch erstelltes Mosaik

Im folgenden werden die berechneten Winkel mit den aus den Kalibrierungsdaten bekannten Winkeln verglichen. In Abbildung 6.19 sind jeweils die absoluten und die relativen Winkel eingetragen. Für die Untersuchung wurden 100 künstliche Bilder aus der Sequenz "Lab" verwendet.

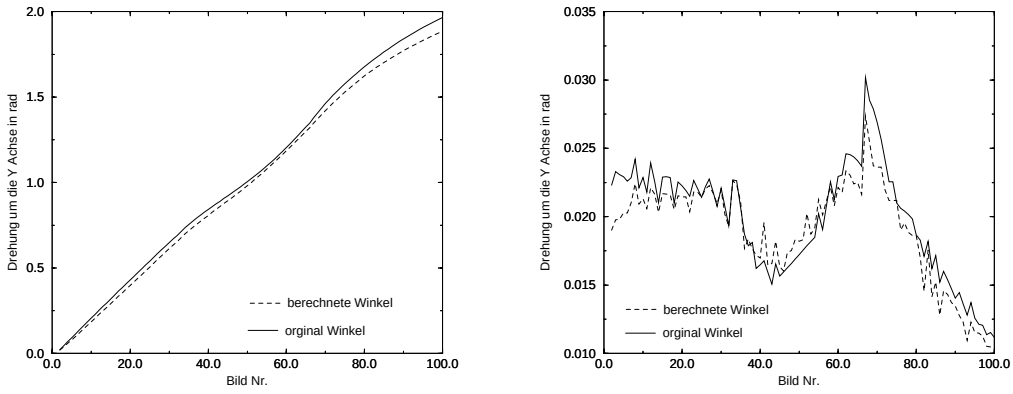



Abbildung 6.19: Genauigkeit der Drehung um die $Y$-Achse; (a) relative Winkel und (b) absolute Winkel

Die relativen Winkel besitzen eine ausreichende Genauigkeit und wurden mit dem Streifenansatz und dem Skalierungsfaktor 8 berechnet. In Tabelle 6.12 werden die durchschnittlichen und die maximalen Fehler für beide Methoden gegenübergestellt.
Die maximalen Fehler fallen mit einem halben Grad bei über 100 Bildern sehr gering aus.

| | |
|---|---|
| Durchschnittlicher Fehler für $X$ Winkel | 0.097 Grad |
| Durchschnittlicher Fehler für $Y$ Winkel | 0.106 Grad |
| Maximaler Fehler für $X$ Winkel | 0.418 Grad |
| Maximaler Fehler für $Y$ Winkel | 0.571 Grad |

Tabelle 6.12: Winkelfehler für ein künstliches Video mit 100 Bildern

Sie wurden in diesem Versuch mit einem Skalierungsfaktor von 10 zurückgerechnet.
Wie anhand Abbildung 6.20 veranschaulicht wird, tritt bei der Datenreduktion durch eine einfachen Skalierung im Vergleich zur Streifen-Methode deutlich ein höherer Qualitätsverlust ein. Des weiteren ist das Skalierungsverfahren weniger robust, da einzelne Pixel größere Auswirkungen haben.

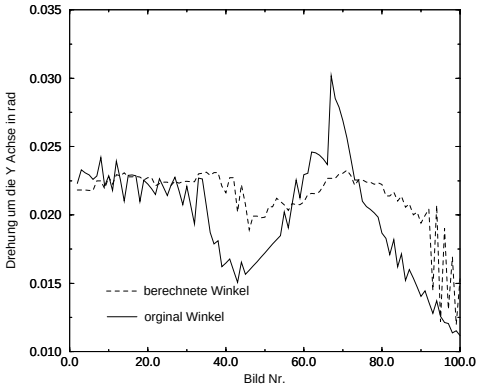



Abbildung 6.20: Winkelgenauigkeit mit der Skalierung

### Geschwindigkeit

Die erreichte Geschwindigkeit wurde anhand von Bildsequenzen ermittelt. Bei der gemessenen Verarbeitungszeit wurde zwischen der Datenreduktion und der Minimierung unterschieden. Die Zeiten für das Laden der Bilder und die Erstellung des Bildmosaiks wurden nicht mitgemessen, da sie für die spätere Echtzeitanwendung nicht von Interesse sind. Das Erzeugen eines Bildmosaiks wird nur für eine visuelle Kontrolle der erzielten Ergebnisse durchgeführt. Die Zeiten sind in User-Time und auf einer SGI O2 mit 180MHz gemessen. Die gewonnenen Ergebnisse sind in Tabelle 6.13 zu sehen. Für die Datenreduktion mit Streifen sind immer die Zeiten des ersten und des zweiten Bild eingetragen. Für das erste Bild müssen die Streifen nur für den mittleren Ausschnitt berechnet werden, wodurch die Berechnungszeiten in etwa halbiert sind. Bei der Bestimmung der Parameter müssen zwei Minimierungen durchgeführt werden.
Es ist deutlich zu erkennen, dass die Zeiten für die Optimierung bei der Verwendung eines einfachen Zooms zu hoch für eine Echtzeitanwendung sind. Die Zeiten der Streifenmethode liegen im Gegensatz zu der Datenreduktion aber schon in einer akzeptablen Größenordnung.

| Bild | Größe in Pixel | Skalierungs-faktor | Methode | t(Datenreduktion) in Sekunden | t(Minimierung) in Sekunden |
|---|---|---|---|---|---|
| Lab | 600x600 | 5 | Streifen | 0.26 0.47 | 0.02 0.008 |
| Lab | 600x600 | 10 | Streifen | 0.23 0.44 | 0.01 0.002 |
| Lab | 600x600 | 15 | Streifen | 0.25 0.44 | 0.008 0.0035 |
| Lab | 600x600 | 5 | Zoom | 0.48 | 1.2 |
| Lab | 600x600 | 10 | Zoom | 0.49 | 0.7 |
| Lab | 600x600 | 15 | Zoom | 0.44 | 0.14 |
| Door | 768x576 | 5 | Streifen | 0.3 0.55 | 0.024 0.023 |
| Door | 768x576 | 10 | Streifen | 0.29 0.54 | 0.01 0.002 |
| Door | 768x576 | 15 | Streifen | 0.3 0.55 | 0.009 0.007 |
| Door | 768x576 | 5 | Zoom | 0.62 | 0.3 |
| Door | 768x576 | 10 | Zoom | 0.59 | 0.46 |
| Door | 768x576 | 15 | Zoom | 0.51 | 0.06 |
| Expo | 720x288 | 5 | Streifen | 0.16 0.27 | 0.01 0.01 |
| Expo | 720x288 | 10 | Streifen | 0.14 0.27 | 0.006 0.005 |
| Expo | 720x288 | 15 | Streifen | 0.16 0.25 | 0.01 0.003 |
| Expo | 720x288 | 5 | Zoom | 0.28 | 0.18 |
| Expo | 720x288 | 10 | Zoom | 0.25 | 0.03 |
| Expo | 720x288 | 15 | Zoom | 0.24 | 0.001 |

Tabelle 6.13: Berechnungszeiten

### 6.3.11 Panoramamosaike

Um einen Eindruck von der Qualität der Ergebnisse zu erhalten, werden in diesem Abschnitt automatisch erstellte Bildmosaike dargestellt. Bei der Erstellung der Mosaike wurden die Bilder nicht deckend überlagert, sondern ihre Pixelwerte wurden aufsummiert. Die Bilder verlieren dabei zwar an Schärfe, aber Helligkeitsunterschiede zwischen den einzelnen Bildern werden unterdrückt.

Abbildung 6.21: Einzelbilder, die für die Erstellung von Abbildung 6.22 genutzt wurden

In Abbildung 6.21 sind drei Ansichten aus der Bildfolge, die zur Erstellung des Mosaiks in Abbildung 6.22 genutzt wurden, dargestellt.
Die Bewegung zwischen den Bildern wurde dadurch hauptsächlich auf eine Drehung um die $Y$-Achse beschränkt. Die Drehungen um die $X$- und die $Y$-Achse wurden bestimmt, wobei die Winkel für die $X$-Achse ungefähr den Wert 0 annehmen. Aus der Originalvideosequenz wurde nur jedes 10. Bild verwendet, da auf Grund einer langsamen Drehung nur kleine

Winkelveränderungen zwischen den Bildern vorliegen. Insgesamt wurden 15 Bilder für die Erstellung des Mosaiks genutzt.

Abbildung 6.22: Mosaik aus 15 Bildern bei einer Kameradrehung um die $Y$-Achse

In dem Mosaik sind optisch keine Fehler oder keinen Übergang zwischen den einzelnen Bildern erkennbar. Abbildung 6.23 zeigt ein Mosaik des Fraunhofer Instituts für Graphische Datenverarbeitung. Das Mosaik ist aus 55 Bildern entstanden, wobei die Kamerabewegung links unten begann und dann in einem Bogen das ganze Gebäude erfasst.

Abbildung 6.23: Fraunhofer Institut für Graphische Datenverarbeitung

### 6.3.12 Zusammenfassung der Ergebnisse

Wie in den Abbildungen 6.22 und 6.23 veranschaulicht, können mit der vorgestellten Methode automatisch Panoramabilder aus Bildfolgen erstellt werden. Um die Echtzeitfähigkeit zu erzielen, ist eine starke Zusammenfassung der Daten, die auf der Aufsummierung der Pixel in Streifen basiert, erforderlich. Die Anwendung des Verfahrens wird jedoch eingeschränkt, da der Überlappungsbereich zweier Bilder relativ groß, d.h. ca. 70% des Bildgröße, betragen muss.
Dieses Verfahren ist von daher zu anfällig, um eine Realisierung des markerlosen Trackingkonzepts, das in Abschnitt 6.1.2 vorgestellt wurde, zu ermöglichen. Ein Alternativverfahren, das auf die Fouriertransformation der Bilder basiert, wurde deswegen untersucht und wird im folgenden vorgestellt.

## 6.4 Fourierbasierte Bildregistrierung

### 6.4.1 Translation

Die Translation zweier Bilder kann mit dem bekannten Verfahren der Phasen-Korrelation präzise bestimmt werden. Die Phasen-Korrelation basiert auf dem Verschiebungstheorem der Fouriertransformation.

#### Verschiebungstheorem für zwei Bilder $f_1$ und $f_2$

Sind zwei Bilder $f_1$ und $f_2$ gegeben, die sich nur durch eine Translation $(t_x, t_y)$ unterscheiden, dann gilt die folgende Gleichung:

$$f_2(x, y) = f_1(x - t_x, y - t_y) \tag{6.10}$$

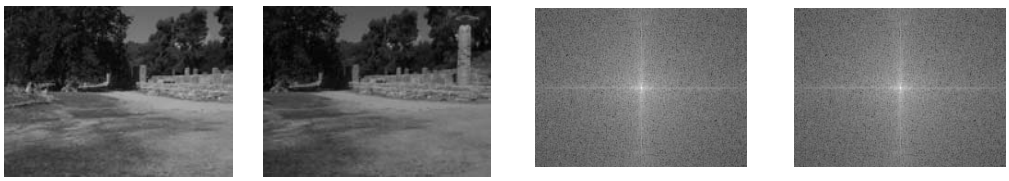
Durch das Verschiebungstheorem der Fouriertransformation gilt für die Fouriertransformierten $F_1$ und $F_2$ der beiden Bilder $f_1$ und $f_2$ folgende Gleichung:

$$F_2(\xi, \eta) = e^{-j2\pi(\xi t_x + \eta t_y)} F_1(\xi, \eta) \tag{6.11}$$

$F_2$ und $F_1$ sind zwei komplexe Matrizen.
Das Kreuz-Leistungsspektrum (cross power spectrum) der fouriertransformierten Bilder $F_1$ und $F_2$ ist wie folgt definiert:

$$\frac{F_1(\xi, \eta) F_2^*(\xi, \eta)}{|F_1(\xi, \eta) F_2^*(\xi, \eta)|} = e^{j2\pi(\xi t_x + \eta t_y)} \tag{6.12}$$

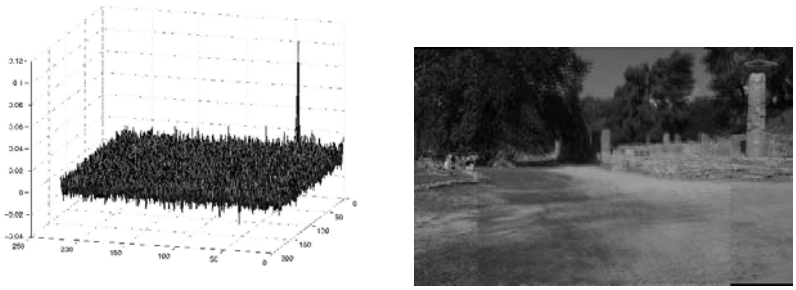
$F_2^*(\xi, \eta)$ stellt den konjugiert-komplexen Wert von $F_2(\xi, \eta)$ dar.
Die Translation $(t_x, t_y)$ ist hiermit nur in der exponentialen Funktion enthalten. Die inverse Fouriertransformation dieser Funktion, ergibt eine Nadelimpuls-Funktion, die überall, außer an der Stelle $(t_x, t_y)$, gleich Null ist. Dies bedeutet, dass durch die Lokalisierung des Nadelimpulses die zwei Komponenten der Translation bestimmt werden können.



(a) (b) (c) (d)

Abbildung 6.24: Kreuzleistungsspektrum zwei zu registrierenden Bildern

Abbildung 6.24 zeigt die verschiedenen Schritte zur Bildregistrierung mit Hilfe der Phasenkorrelation. In Abbildung 6.24(a) und (b) sind die Originalbilder und deren Amplitudenspektrum in Abbildung 6.24(c) und (d) zu sehen.



(e) (f)

Abbildung 6.25: Nadelimplus vom Kreuzleistungsspektrum

Die Inverstransformation des Kreuzleistungsspektrums ergibt einen klar definierten Nadelimpuls, in Abbildung 6.24(e), an der Stelle der Verschiebungswerte $(t_x, t_y)$. Die registrierten Bilder sind in Abbildung 6.24(f) zu sehen.

### 6.4.2 Translation und Rotation

Wenn das Bild $f_2(x, y)$ eine rotierte und verschobene Kopie von $f_1(x, y)$ mit der Verschiebung $(t_x, t_y)$ und dem Winkel $\phi$ ist, gilt für $f_1$ und $f_2$:

$$f_2(x, y) = f_1(x \cos\phi_0 + y \sin\phi_0 - t_x, -x \sin\phi_0 + y \cos\phi_0 - t_y) \tag{6.13}$$

Die Fouriertransformierten $F_1$ und $F_2$ von $f_1$ und $f_2$ sind wie folgt definiert:

$$F_2(\xi, \eta) = e^{-j2\pi(\xi t_x + \eta t_y)} F_1(\xi \cos\phi_0 + \eta \sin\phi_0, -\xi \sin\phi_0 + \eta \cos\phi_0) \tag{6.14}$$

Die Amplituden $M_1$ und $M_2$ der komplexen Funktionen $F_1$ und $F_2$, auch Amplitudenspektren von $f_1$ und $f_2$ genannt, werden anschließend berechnet. Da die Amplitude der Exponentialfunktion eins beträgt, gilt folgende Gleichung:

$$M_2(\xi, \eta) = M_1(\xi \cos\phi_0 + \eta \sin\phi_0, -\xi \sin\phi_0 + \eta \cos\phi_0) \tag{6.15}$$

Daraus ergibt sich, dass die Transformation zwischen den zwei Amplitudenspektren nur der Rotation, die die ursprünglichen Bilder unterscheidet, entspricht. Die Amplitudenspektren sind unabhängig von der Translation.
Der Rotationswinkel kann mit Hilfe einer Transformation von $M_1$ und $M_2$ in ein Polar-Koordinatensystem gut erfassen werden. Ein Punkt $P(\xi, \eta)$ wird von den Amplitudenspektren durch den Punkt $P(r, \phi)$ repräsentiert. Die Gleichung 6.15 kann dann wie folgt geschrieben werden:

$$M_2(r, \phi) = M_1(r, \phi - \phi_0) \tag{6.16}$$

Die obige Gleichung zeigt, dass die Rotation der Amplitudenspektren durch eine Verschiebung $(0, \phi_0)$ im Polarkoordinatensystem ausgedrückt wird. Diese Verschiebung und dadurch der Rotationswinkel $\phi_0$ können auf einfache Weise mit der Phasen-Korrelationsmethode, die im vorherigen Abschnitt erläutert wurde, bestimmt werden.
Nach der Erfassung des Rotationswinkels wird das zweite Bild um den Betrag $\phi_0$ rücktransformiert. Die Transformation vom Bild $f_1$ zum Bild $f_2$ besteht dann nur aus der Verschiebung, die mit der Phasen-Korrelationsmethode berechnet wird.

### 6.4.3 Skalierung

Die Skalierung zwischen zwei Bildern kann ähnlich wie im Abschnitt "Translation und Rotation" gefunden werden. Wenn das Bild $f_2(x, y)$ eine skalierte Kopie von $f_1(x, y)$ mit den Skalierungsfaktoren $(a, b)$ ist, gilt:

$$f_2(x, y) = f_1(ax, by) \tag{6.17}$$

Durch die Fouriertransformation erhält man:

$$F_2(\xi, \eta) = \frac{1}{|ab|} F_1\left(\frac{\xi}{a}, \frac{\eta}{b}\right) \tag{6.18}$$

In diesem Fall wird eine logarithmische Transformation auf $F_1$ und $F_2$ angewendet, um die Skalierungsfaktoren zu erfassen. Daraus folgt aus Gleichung 6.18:

$$F_2(\log \xi, \log \eta) = \frac{1}{|ab|} F_1(\log \xi - \log a, \log \eta - \log b) \tag{6.19}$$

Die Skalierung kann hiermit als eine Verschiebung der Funktionen $F_1$ und $F_2$ ausgedruckt werden. Mit Hilfe des Phasen-Korrelationsverfahrens können $\log(a)$ und $\log(b)$ und anschließend die Skalierungsfaktoren berechnet werden.

### 6.4.4 Skalierung und Rotation

In der Praxis kann davon ausgegangen werden, dass die Skalierungen in horizontaler und vertikaler Richtung, so wie beispielsweise im Fall eines Kamerazooms, gleich groß sind.
Unter dieser Voraussetzung kann eine gleichzeitige Rotation und Skalierung eines Bildes durch Transformation der Amplitudenspektren in Log-Polar-Koordinaten bestimmt werden, vergleiche auch Gleichung 6.16. Nach dieser Konvertierung folgt für die Amplitudenspektren die Gleichung:

$$M_2(\log r, \phi) = M_1(\log r - \log a, \phi - \phi_0) \tag{6.20}$$

Die Verschiebungen entlang der X-Achse und der Y-Achse entsprechen jeweils der Skalierung $\log a$ und dem Rotationswinkel $\phi_0$.

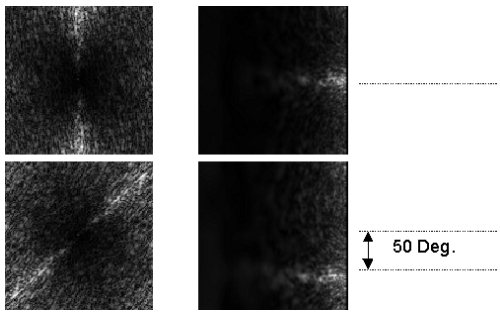



Abbildung 6.26: Amplitudespektren und Log-Polar-Darstellung (Drehung um 50 Grad)

Die Log-Polar-Transformation wird Abbildung 6.28 für zwei um 50 Grad gedrehte Amplitudespektren dargestellt.

### 6.4.5 Fouriertransformation digitaler Bilder

#### Die Fast-Fouriertransformation (FFT)

Eine sehr bekannte und effiziente Implementierungsmethode der Fouriertransformation stellt die sogenannte *Fast-Fouriertransformation* (FFT) dar. In dieser Arbeit wurde die FFT-Implementierung, die in [69] beschrieben wird, verwendet.
Die FFT setzt ein quadratisches Bildformat, dessen Dimension $2^n$ betragen muss, voraus. Entspricht das Format des Bildes diesen Anforderungen nicht, muss ein quadratischer Bildausschnitt gewählt werden, der dann auf die nächstgrößte Dimension $2^n$ runter- oder hochskaliert wird.

#### Log-Skalierung der Grauwerte

Für die Praxis relevant sind das Amplitudenspektrum $|F(\xi,\eta)|$ und das Phasenspektrum $\Phi(\xi,\eta)$. Mit zunehmender Frequenz werden die komplexen Koeffizienten der Fourierspektren sehr klein. Bei linearer Skalierung in einen Grauwertebereich $[0,255]$ gehen sehr viele Informationen verloren. Deshalb wird meist eine logarithmische Skalierung der Grauwerte eines Amplitudenspektrums gewählt. In Formel 6.21 ist eine Möglichkeit zur logarithmischen Skalierung der Koeffizienten des Amplitudenspektrums angegeben. Somit kann eine einfache Grauwert-Transformation der Amplitudenwerte erfolgen.

$$|F(\xi,\eta)| = \log\left(1+\sqrt{Real^2(\xi,\eta)+Imag^2(\xi,\eta)}\right) \tag{6.21}$$

Um eine anschauliche Darstellung des Amplitudenspektrums zu erhalten wurde zusätzlich der Koordinatenursprung in die Bildmitte verschoben.

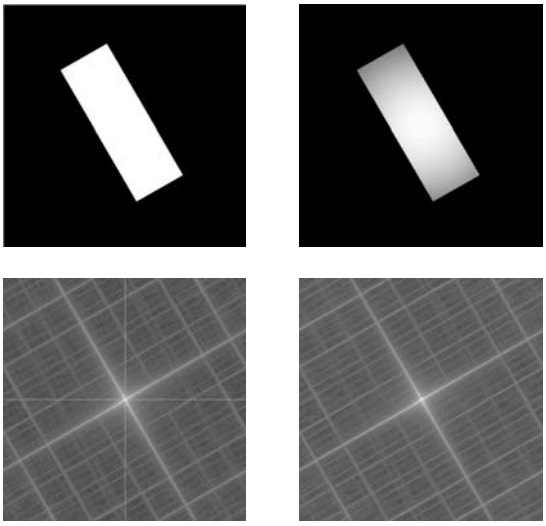
#### Fensterfunktion

Die Fouriertransformation ist für unendliche und periodische Funktionen definiert. Bei endlichen Funktionen besteht die Periodizität aus der Fortsetzung dieses Signals. Dies verursacht Diskontinuitäten am Rand des Definitionsbereichs der Funktion. Dadurch erscheinen neue Frequenzen, die in der ursprünglichen Funktion nicht vorhanden sind und zu einer Verfälschung des Spektrums führt.
Deswegen soll das Bild mit einer Fensterfunktion, die einen kontinuierlichen Abfall der Pixelwerte am Rand gewährleistet, multipliziert werden. Dieses Problem wird in Abbildung 6.27 veranschaulicht. Im ersten Spektrum (a) erscheint auf Grund der Diskontinuitäten ein Kreuz, das nach der Multiplikation des Eingabebildes mit einer Sinus-Fensterfunktion im Spektrum (b) nicht mehr vorhanden ist.

#### Log-Polar Darstellung

Um zwei Bilder registrieren zu können, ist es notwendig, die Amplitudenspektren der beiden Bilder in Log-Polar-Darstellung zu konvertieren.

In Abbildung 6.28 ist schematisch dargestellt, wie die kartesischen Koordinaten in das Log-Polar-Koordinatensystem überführt werden. Anstatt der Koordinaten $(\xi,\eta)^\top$ wird nun der logarithmierte Abstand vom Koordinatenursprung und der Winkel zur positiven



(a) (b)

Abbildung 6.27: (a) Eingabebild und Leistungsspektrum (c) Eingabebild mit Fensterfunktion und Leistungsspektrum

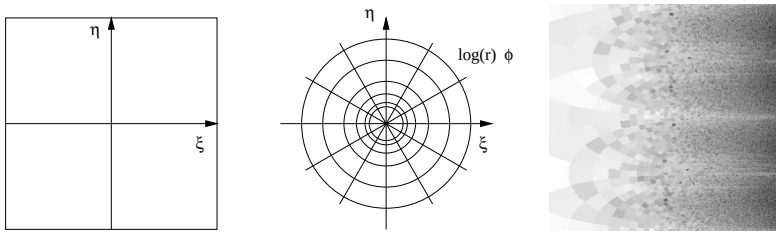



Abbildung 6.28: Konvertierung zum Log-Polar Koordinatensystem

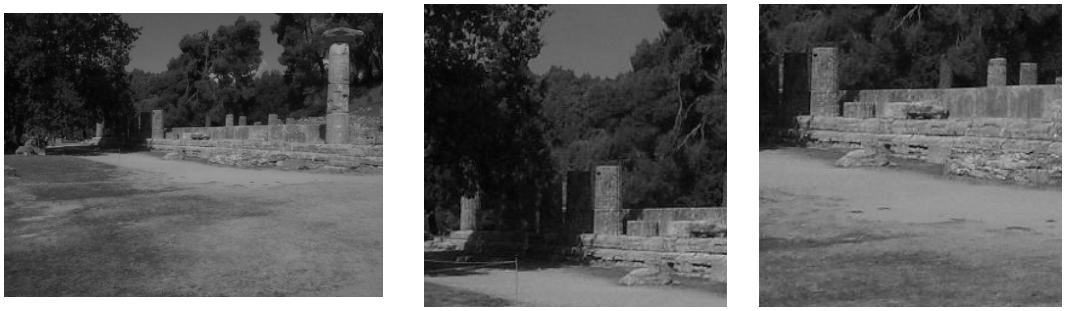
$X$-Achse verwendet $(\log r, \phi)$. Auf der rechten Seite der Abbildung wird ein nach Log-Polar-Koordinaten transformiertes Amplitudenspektrum gezeigt.

### 6.4.6 Evaluierung des Verfahrens

Zunächst wird das Verfahren an Bildern, die sich durch eine bekannte euklidischen Transformation (Translation, Rotation, Skalierung) unterscheiden, getestet.

Aus einem hochaufgelösten Bild werden zwei Bildauszüge mit der Größe $256 \times 256$ Pixels geschnitten, siehe Abbildung 6.29. Die Transformation zwischen diesen Ausschnitten ist gegeben und wird mit Hilfe des FFT-Verfahrens zurückgerechnet. Damit können die berechneten Werten mit den bekannten richtigen Werten ("Ground Truth") verglichen und die Genauigkeit des Verfahrens, für z.B. verschiedene Rauschpegel oder unterschiedlich große Überlappungsbereiche, evaluiert werden. Insbesondere werden die folgenden Themen behandelt:

- Evaluierung des notwendigen minimalen Überlappungsbereiches zwischen den Bildern,



(a) (b) (c)

Abbildung 6.29: OriginalBild (a) und berechnete Bildausschnitte (b,c)

- Definition eines Gültigkeitskriteriums, das eine erfolgreiche Registrierung charakterisiert,
- Einfluss des Bildrauschens. Hierfür wird ein weißes Rauschen mit verschiedenen Stärken ($\sigma = 0, 50, 100, \ldots, 600$) zu den Bildausschnitten addiert.

**Translation**

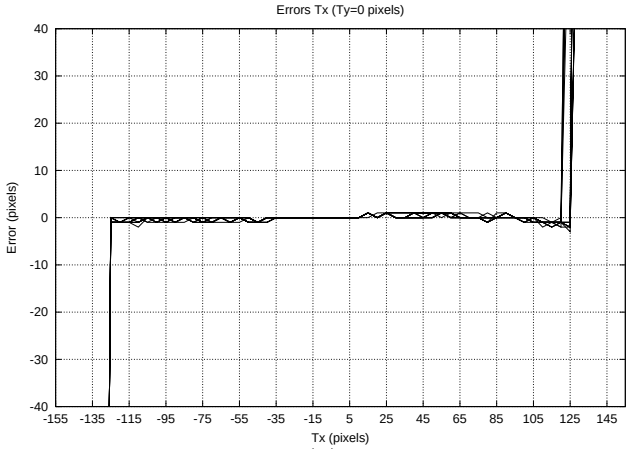
Die ersten Untersuchungen evaluieren das Verfahren für zwei Bilder, die sich nur durch eine Verschiebung $(T_x, T_y)$ unterscheiden. Die Translation entlang der $X$-Achse $(T_x)$ variiert von -135 bis 135 Pixel in 5 Pixel-Schritten für die vorgegebenen Verschiebungen $T_y$ ($T_y = -105, -90, -75, 0, 75, 90, 105$). Dazu werden die Bildregistrierungsversuche für zwölf vordefinierten Rauschpegel, nämlich für $\sigma = 0, 50, 100, \ldots, 600$, wiederholt.
Die Bildtransformation wird mit dem FFT-Verfahren zurückgerechnet, wobei alle Parameter (Rotationswinkel, Skalierung, Verschiebung) und nicht nur die Verschiebung zugelassen werden. Die Bilder in Abbildung 6.30 veranschaulichen diese Vorgehensweise. Die Bilder (a) und (b) stellen zwei FFT-Registrierungsergebnisse für eine Verschiebung ($(T_x, T_y) = (115, 105)$) zwischen den Bildausschnitten. Die Bilder in (a) wurden nicht gestört, während in (b) ein Rauschen mit $\sigma = 600$ addiert wurde.

(a) (b)

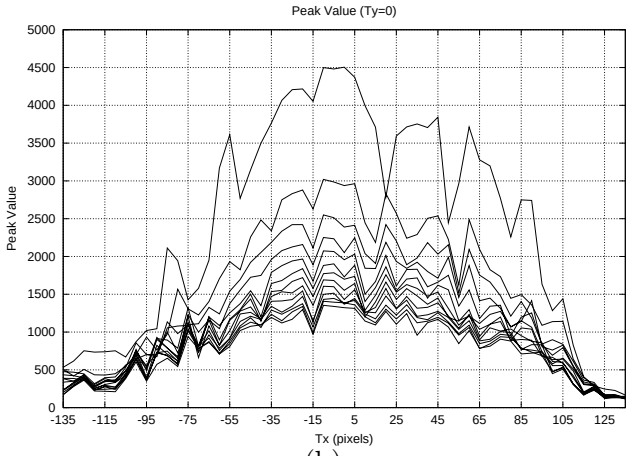
Abbildung 6.30: Registrierungsbeispiel zweier Bildausschnitte, ohne künstliches Rauschen (a) und mit künstlichem Rauschen (b)

In Abbildung 6.31(a) sind die Fehler, die bei der Berechnung von $T_x$ mit $T_y = 0$ für

alle Rauschensstärken verursacht wurden, zusammengestellt. Eine erfolgreiche Registrierung wird für den Bereich $T_x = [-125; 125]$ unabhängig von der Rauschstärke erreicht. Die Fehler betragen ein bis zwei Pixel und der minimale Überlappungsbereich entspricht 51% der gesamten Bildfläche. Der Einfluss des Rauschens kann erst nach der Analyse der Werte des Nadelimpulses, in Abbildung 6.31(b) dargestellt, erfasst werden. Für eine gegebene Verschiebung sinkt der Impulswert bei steigendem Rauschen. Für $T_x = 5$ variiert der Impulswert beispielsweise von 4500 bis zu 1400. Das Flacherwerden der Peak-Kurve bedeutet, dass die Unterscheidung zwischen einer erfolgreichen und einer fehlenden Registrierung schwieriger und unsicherer wird.



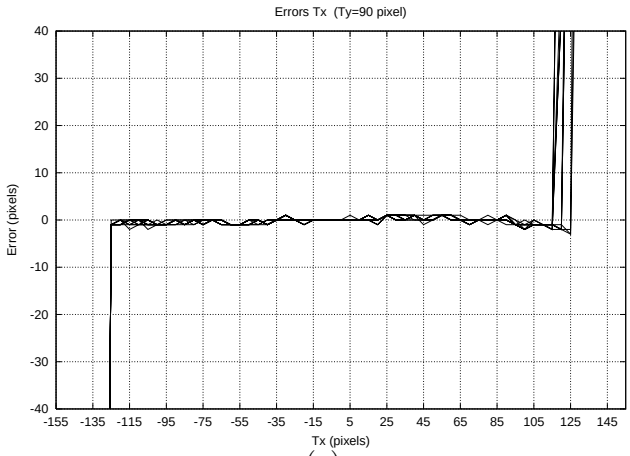

(a)



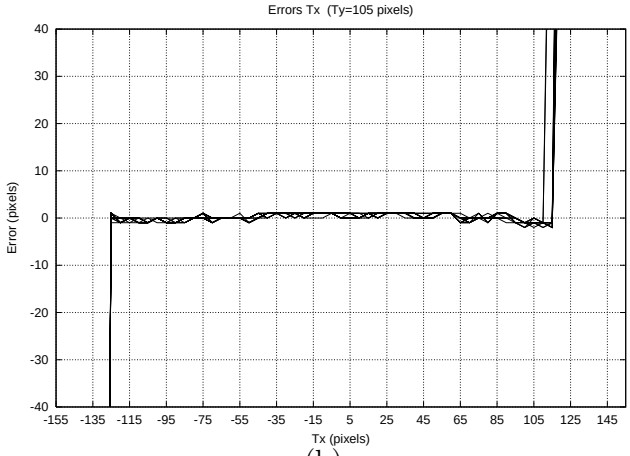

(b)

Abbildung 6.31: Berechnungsfehler von $T_x$ ($T_y = 0$)(a) und entsprechende Werte des Nadelimpulses (b)

Für die weiteren Verschiebungen $T_y = 75, 90, 105$ kann ein ähnliches Verhalten des Verfahrens beobachtet werden. Die Fehler behalten eine Größenordnung von 1 bis 2 Pixeln und der minimale Überlappungsbereich sinkt in einigen Fällen bis zur 32% ($T_y = 105; T_x = 115$), siehe Abbildung 6.32.
Die Ergebnisse, wie in Abbildung 6.33 dargestellt, für eine negative Verschiebung in $Y$-



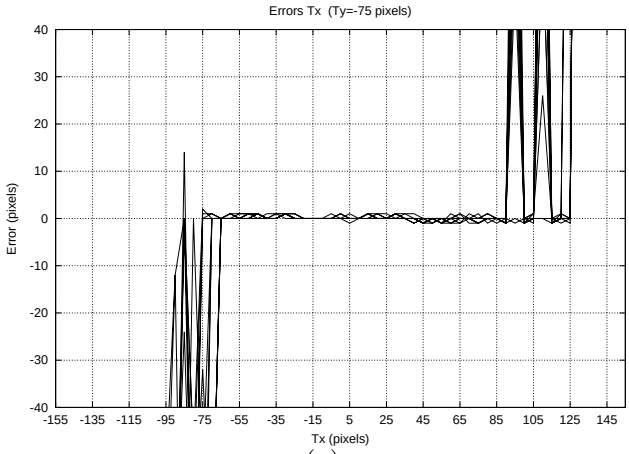

(a)



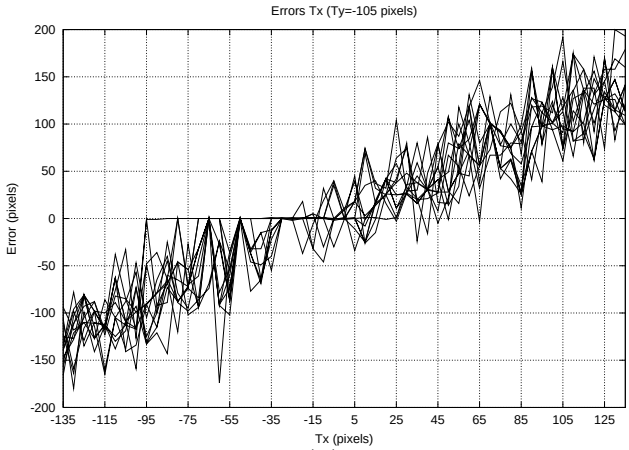

(b)

Abbildung 6.32: Berechungsfehler von $T_x$ mit $T_y = 75$ (a) und $T_y = 105$ (b)

Richtung sind wesentlich schlechter. Dabei wird für $T_y = -75$ der Bereich einer gültigen Registrierung wesentlich kleiner. Für $T_y = -105$ können nur Bilder ohne Rauschen registriert werden, siehe Abbildung 6.33. Das Verfahren reagiert hier extrem empfindlich. Dieses Problem spiegelt sich auf die Werte des Nadelimpulses, in Abbildung 6.34 dargestellt, deutlich wieder. Die Impulswerte sind insgesamt viel niedriger als für $T_y$ positiv, und sinken sehr schnell ab.




(a)




(b)

Abbildung 6.33: Berechungsfehler von $T_x$ mit $T_y = -75$ (a) und $T_y = -105$ (b))

Die Erklärung dafür ist auf dem Bildinhalt zurückzuführen. Für $T_y < 0$ werden die Bildausschnitten aus dem unteren Bereich des ursprünglichen Bildes genommen, siehe Abbildung 6.30. Dieser Bereich ist wenig strukturiert und besteht aus einer uniformen Region, dem Sandboden. Im Gegensatz dazu beinhalten die Bildausschnitte für $T_y > 0$ unterschiedliche, eindeutige Regionen, Steine des Tempels, Bäume, Himmel, die die Bildregistrierung unterstützen und sehr gute Ergebnissen trotz hoher Rauschwerte ermöglichen.

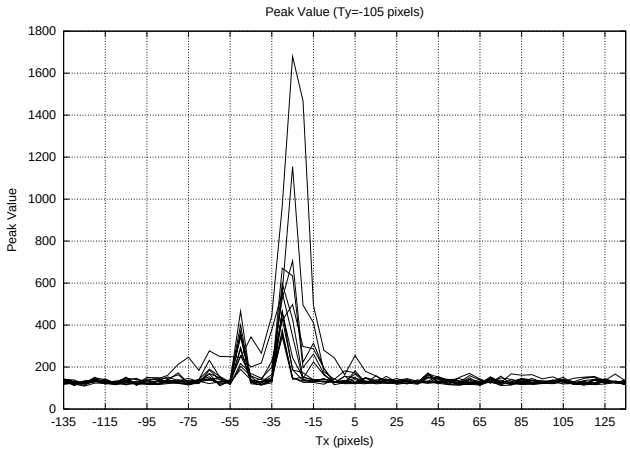



Abbildung 6.34: Berechungsfehler von $T_x$ mit $T_y = -75$ (a) und $T_y = -105$ (b)

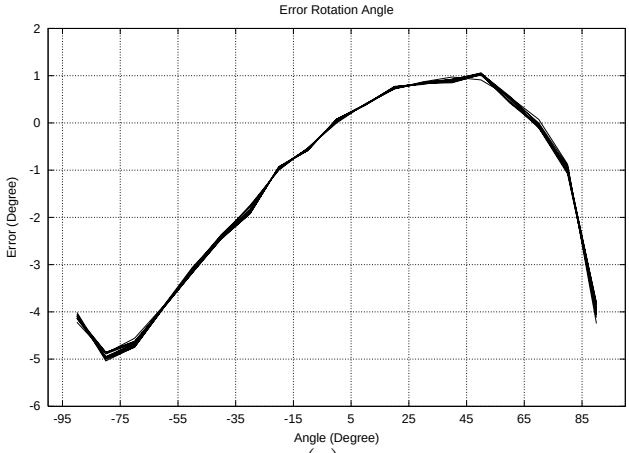
### Rotation

In diesem Abschnitt wird zunächst die Berechnung des Rotationswinkels untersucht. Hierfür werden die Bildausschnitte in ihren Zentren von $-90$ bis 90 Grad gedreht, und, wie für die Translation, der berechnete Winkel mit dem ursprünglichen Winkel verglichen. Abbildung 6.35 zeigt zwei Bildausschnitte für einen Rotationswinkel von 50 Grad und das entsprechende Registrierungsergebnis.

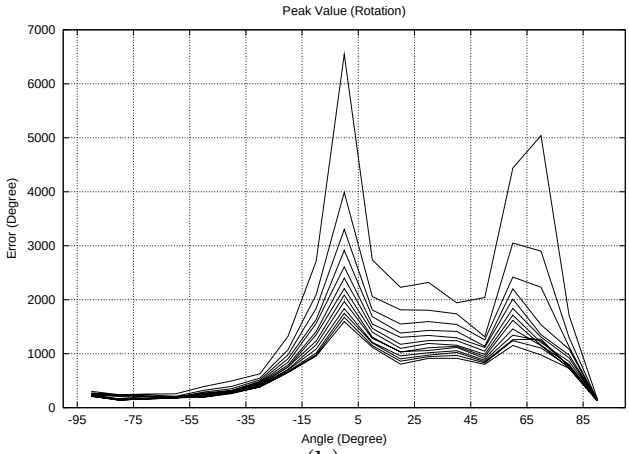
(a) (b) (c)

Abbildung 6.35: Registrierung zweier um 50 Grad gedrehte Bilder

Die Winkelfehler sind allgemein relativ klein und bleiben unter 1 Grad für Rotationswinkel von -20 bis zu 80 Grad. Dieses Ergebnis spiegelt sich in den Werten des Nadelimpuls wieder, siehe Abbildung 6.36. Kleine Fehler stimmen mit hohen Peak-Werten überein und können hiermit die Güte der Registrierung charakterisieren.




(a)




(b)

Abbildung 6.36: Berechnungsfehler der Rotationswinkel (a) und entsprechende Werte des Nadelimpulses (b)

### Skalierung

Für die Überprüfung der Registrierungsergebnisse nach einer Skalierung wird ein Ansatz verwendet, der dem Ansatz bei Behandlung einer Rotation ähnelt. Bildausschnitte werden mit Skalierungsfaktoren zwischen 1 bis 2 erzeugt und aufeinander registriert. Ein Beispiel ist in Abbildung 6.37 für einen Skalierungsfaktor von 1.7 veranschaulicht.

(a) (b) (c)

Abbildung 6.37: Registrierungsbeispiel mit einer Skalierung $S = 1.7$

Die Skalierungsfehler und die Werte des Nadelimpulses werden in Abbildung 6.38 (a) bzw. (b) dargestellt. Das Verfahren liefert bis zur einer Skalierung von 1.85 gute Ergebnisse. Für die Skalierung kann ein ähnlicher Einfluss des Rauschens, wie bei den vorherigen Untersuchungen beobachtet werden.

### Untersuchungen mit Videosequenzen

In Abbildung 6.39 sind Bilder einer Videosequenz, bei der die Kamera um die Stativachsen gedreht wurde, dargestellt.
Die Bilder sind teilweise unterschiedlich beleuchtet und an Kanten tritt der Kammeffekt unterschiedlich stark auf. Die Ergebnisse sind insgesamt bemerkenswert, da zwischen diesen Bildern nur ein kleiner überlappender Bereich (ca. 45%) vorhanden ist. Die Registrierung, siehe 6.39(a)+(c), hat nicht funktioniert, da zu wenig Gemeinsamkeit zwischen den Bildern vorliegen.
In Abbildung 6.40 wurde die Kamera in der Hand gehalten und Szeneänderungen vorgenommen, indem einige Stellen der Fahrzeugtür durch den Arm eines Betrachters verdeckt wurden. Der überlappende Bereich der Bilder ist teilweise extrem klein und beträgt bei der Registrierung Abbildung 6.40(b) und (g) etwa 30%. Das Verfahren liefert jedoch trotz Szenenänderungen gute Ergebnisse und verhält sich insgesamt sehr robust.

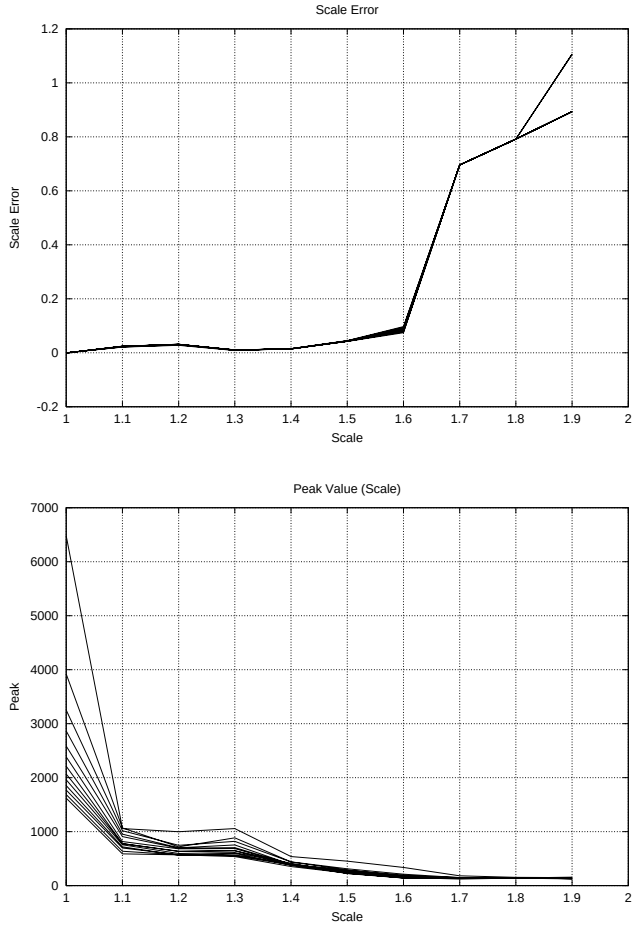



Abbildung 6.38: Berechnungsfehler des Skalierungsfaktors $S$ ($S = 1, 1.1, \ldots, 2$)

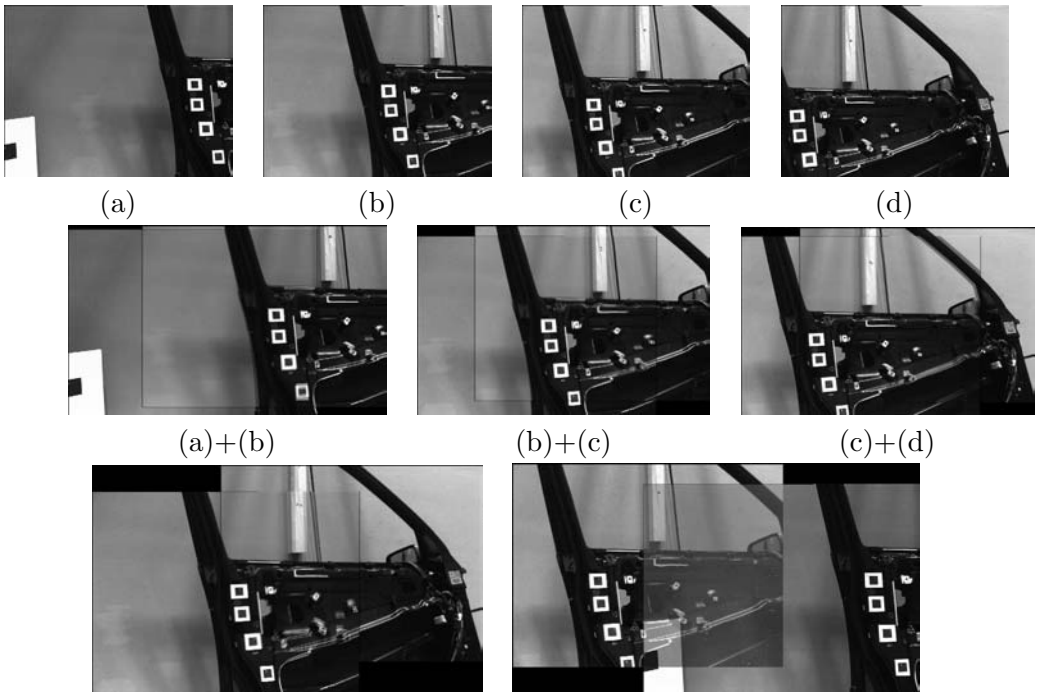



Abbildung 6.39: Bildfolge mit Kameradrehung um Stativachse

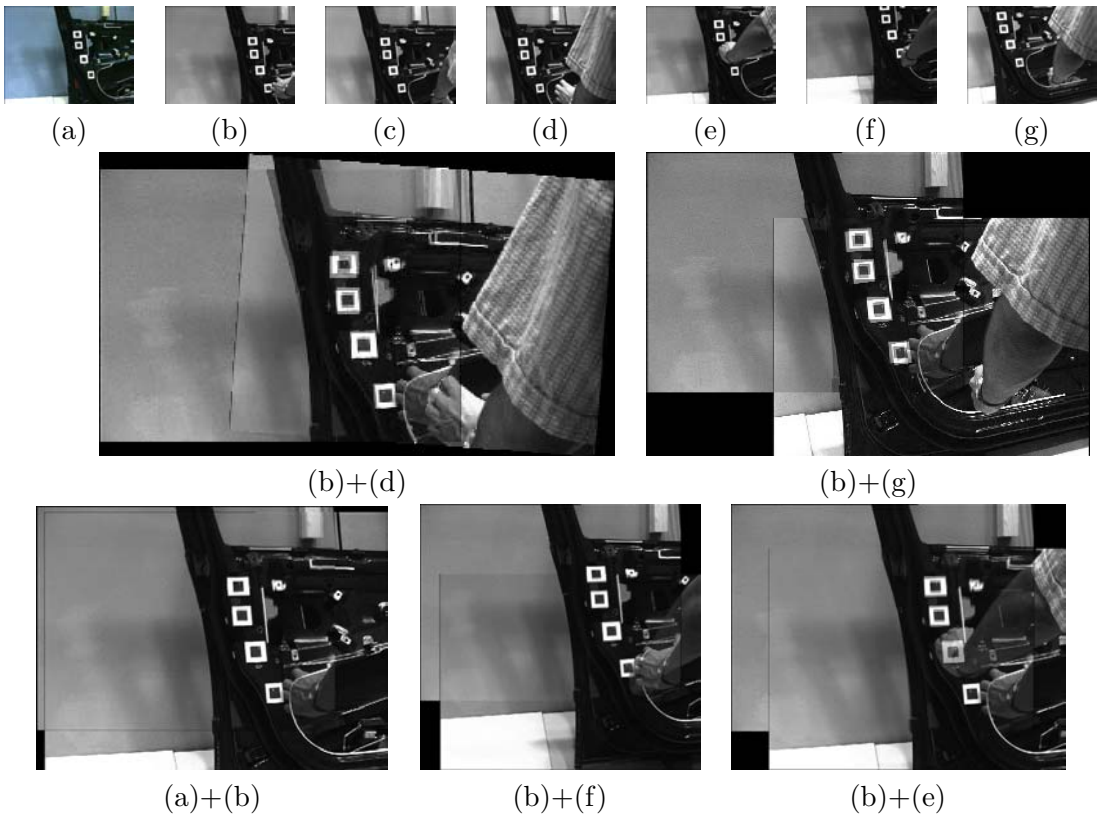
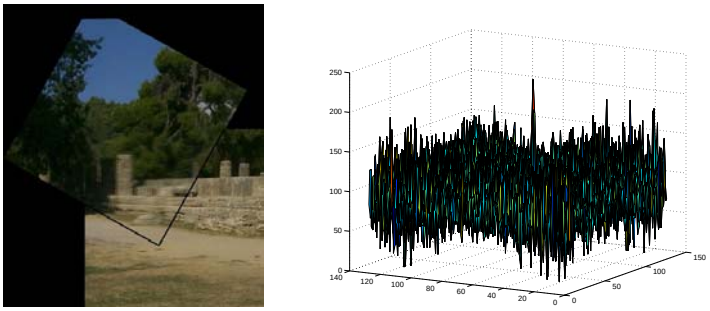



Abbildung 6.40: Registrierung mit freien Drehungen und Szenenänderungen

**Ableitung eines Gütekriteriums der Bildregistrierung**

Die Phasenkorrelation bietet den Vorteil, dass der Nadelimpuls ein Gütekriterium über die Validität der Registrierung liefert. Hierfür wird der Mittelwert $\bar{m_p}$ und die Standardabweichung $\sigma_p$ über alle Werte berechnet und überprüft, ob das Maximum, der Nadelimpuls $N_p$, eindeutig definiert ist. Eine eindeutige Definition des Maximums liegt vor, wenn es sich von dem Mittelwert unterscheidet, d.h. folgender Test: $M_p > k.\sigma_p + \bar{m_p}$ erfüllt wird. Der Faktor $k$ wurde für die Implementierung des Verfahrens auf den Wert drei gesetzt.



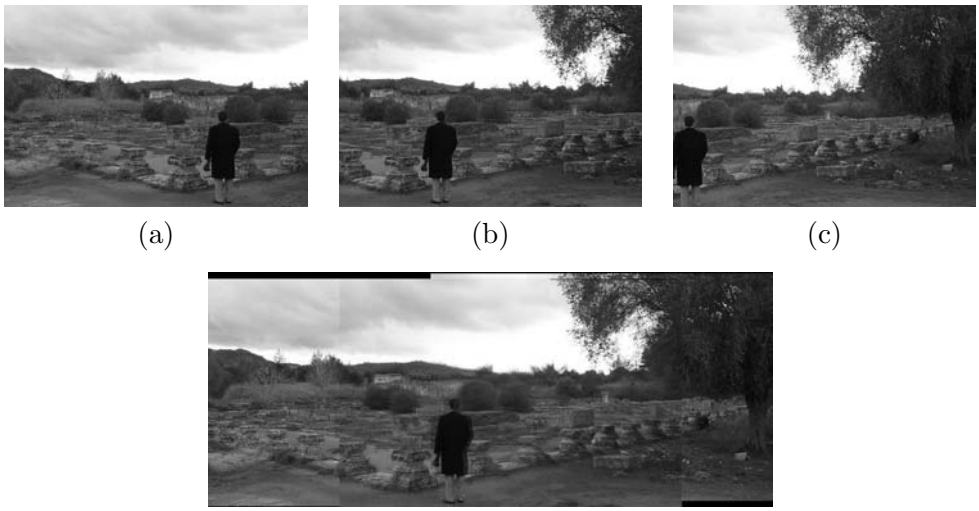

(a) (b)

Abbildung 6.41: Registrierungsbeispiele mit $T_x = 80$, $T_y = 80$ und Rotationswinkel $\phi = 45$ Grad (a); Dreidimensionale Darstellung der Fourier-Inverstransformation (b)

Abbildung 6.41(a) zeigt eine erfolgreiche Bildregistrierung mit einem sehr kleinen Überlappungsbereich. Der Nadelimpuls wurde richtig lokalisiert und liefert die korrekten Komponenten der Rotation und des Verschiebungsvektors zurück. Dennoch zeigt die 3D-Darstellung der Inverstransformation des Kreuzleistungsspektrums wie unsicher dieses Ergebnis ist. Der Nadelimpuls ist von den anderen Werte kaum zu unterscheiden. Ein solches Ergebnis wird durch das vorherige Validitätskriterium ausgeschlossen.

### 6.4.7 Bildung von Bildmosaiken

Weitere Untersuchungen wurden mit einer erhöhten Bildanzahl durchgeführt, wobei aus den Bildern Bildmosaike gewonnen wurden. Das FFT-Registrierungsverfahren liefert auch in diesem Fall trotz problematischer Beleuchtungsverhältnissen sehr gut Ergebnisse der "Out-door"-Szene, siehe Abbildung 6.42.
Das in Abbildung 6.43 dargestellte Bildmosaik wurde auf die gleiche Weise erzeugt.
Abbildung 6.44 zeigt das Ergebnis eines Bildmosaik, das in Echtzeit über eine lange Videosequenz gebildet wurde.
Hierzu wurden die Bildsequenz "Labor" entsprechend Abschnitt 6.3.11 ausgewählt. Die Kamera war bei der Aufnahme auf einem Stativ befestigt und die Rotation erfolgte um die $x-$ und $y-$ Achse. Da das Phasen-Korrelations-Verfahren nur euklidische Transformationen einsetzt und bei der Aufnahme der Bilder kein "Zoom" verwendet wurde, sind die Bilder hauptsächlich aufeinander verschoben worden. Das Ergebnis sieht korrekt aus, da die Objekte in einer großen Entfernung zur Kamera stehen. Dennoch wird bei dem Vergleich mit den Ergebnissen des intensitätsbasierten Verfahrens, siehe Abbildung 6.23, deutlich, dass die perspektivische Verzerrung nicht beachtet wurde.
Das Bildmosaik in Abbildung 6.45 wurde ebenfalls mit der Phasenkorrelationsmethode erzeugt.



(a) (b) (c)

Abbildung 6.42: Bildmosaik (Olympia, ArcheoGuide-Projekt)

Abbildung 6.43: Bildmosaik mit Phasenkorrelationsverfahren

Die zu registrierenden Bilder stammen aus der Videosequenz von Abbildung 6.18 im Abschnitt 6.3.10. Die Objekte (Tische, Stühle) sind hier näher an der Kamera und die Annahme der Registrierung mit einer euklidischen Transformation ist nicht mehr gültig. Zur korrekten Registrierung wird hier eine projektive Transformation benötigt. Das Verfahren ist jedoch robust genug, um die optimale Verschiebung zu finden.

### 6.4.8 Vergleich mit dem intensitätsbasierten Verfahren

Die Ergebnisse des intensitätsbasierten Verfahrens können wie folgt zusammengefasst werden:

1. Das Verfahren erfasst die richtige (projektive) Bildtransformation.
2. Die Echtzeit-Performanz kann nur mit einer minimalen Parametrisierung erreicht werden. In der Praxis werden die Parameter auf zwei begrenzt (Kamera auf einem Stativ).
3. Die Echtzeit-Bildregistrierung ist durch Bildpyramiden und starke Datenreduktion möglich.

Abbildung 6.44: Echtzeit-Tracking mit FFT-basierter Bildregistrierung

Abbildung 6.45: Fehlerhafte Bildung einer Bildmosaik aufgrund 2D-Euklidischen Transformationen und starken Kamerarotationen

4. Muster, die sich im Bild wiederholen, verursachen lokale Minima. Eine automatische Registrierung ist dann oft nicht möglich.

5. Der Überlappungsbereich zwischen den Bildern muss groß sein (etwa 80%). Das Verfahren kann von daher nur für kontinuierliche Bildfolgen eingesetzt werden. Bei kleineren Überlappungsbereichen müssen die Startwerte nah zur endgültigen Lösung stehen.

Das fourierbasierte Verfahren hat sich insgesamt als sehr robust erwiesen. Die wichtigsten Eigenschaften des Verfahrens sind die folgenden:

1. Die Bildtransformation ist eine 2D-euklidische Transformation.

2. Es werden keine Initialisierungswerte benötigt, d.h. die Registrierung erfolgt voll automatisch.

3. Der überlappende Bildbereich kann relativ klein sein. Eine korrekte Registrierung konnte mit einem gemeinsamen Bereich von nur $35 - 50\%$ erzielt werden. Dadurch zeigte das Verfahren auch gute Ergebnisse bei geringen Szeneänderungen.

4. Ein Gütekriterium der Registrierung ist durch den Nadelimpuls des Kreuzleistungsspektrums gegeben.

Da die Zuverlässigkeit und Robustheit des fourierbasierten Verfahrens wesentlich größer ist, wurde es für das markerlose Trackingsystem, im Abschnitt 6.1.2 vorgestellt, ausgewählt.

## 6.5 Bild-Selektion

### 6.5.1 Das Bild-Selektionsmodul

Das Trackingsystem soll, um einen größeren Anwendungsbereich anbieten zu können, auf Basis mehrerer Referenzbilder arbeiten. Da die Bildregistrierung einen aufwendigen Prozess darstellt und dennoch Echtzeit-Tracking auf tragbaren Rechnern erreicht werden soll, wurde ein Pre-Selektionsmobul eingeführt. Die Aufgabe dieses Moduls ist die ähnlichste Ansicht aus einer Menge von mehreren Referenzbildern zu selektieren und für die Bildregistrierung bereit zu stellen.
Dieses Modul muss sehr effizient sein, da die Anzahl der Referenzbilder nicht die Performance des Systems beeinträchtigen soll. Die Anwendung des FFT-Registrierungsverfahrens für das Bild-Selektionsmodul ist ohne weiteres nicht möglich, da jedes zusätzliche Referenzbild eine Halbierung der Echtzeit-Leistung bedeutet. Darüber hinaus wird bei jeder Bildregistrierung die komplette Transformation berechnet, die für die Pre-Selektion nicht notwendig ist. Um eine hohe Verfahrensgeschwindigkeit zu realisieren, sollen in dieser Etappe lediglich die Bildinhalte verglichen werden.
Der Bereich der inhaltsbasierten Bildselektion ist als *Content-based Image Retrieval* in der Literatur bekannt. Das grundlegende Prinzip der verwendeten Methoden ist die Definition und der Vergleich verschiedener Bilddeskriptoren, die aussagekräftige Bildmerkmale darstellen. Dies beinhaltet z.B. die Repräsentation der Bildfarbe über ein Histogramm oder die Extraktion von uniformen Bildregionen. Die statistische und räumliche Beschreibung der Bilddeskriptoren ermöglicht die Bestimmung einer Distanz, die die Ähnlichkeit der Bildinhalte charakterisiert. Die zwei bekanntesten inhaltsbasierten Systeme sind das QBIC- [36] und das VIRAGE-System [1].
Aufgrund der Bildregistrierung stehen schon die Fourierkoeffizienten der Bilder zur Verfügung. Diese beschreiben ebenfalls den Bildinhalt und können daher für die Bildselektion genutzt werden [78]. Eine fourierbasierte Methode weist in diesem Fall den Vorteil auf, daß keine neuen Merkmale bestimmt werden müssen.

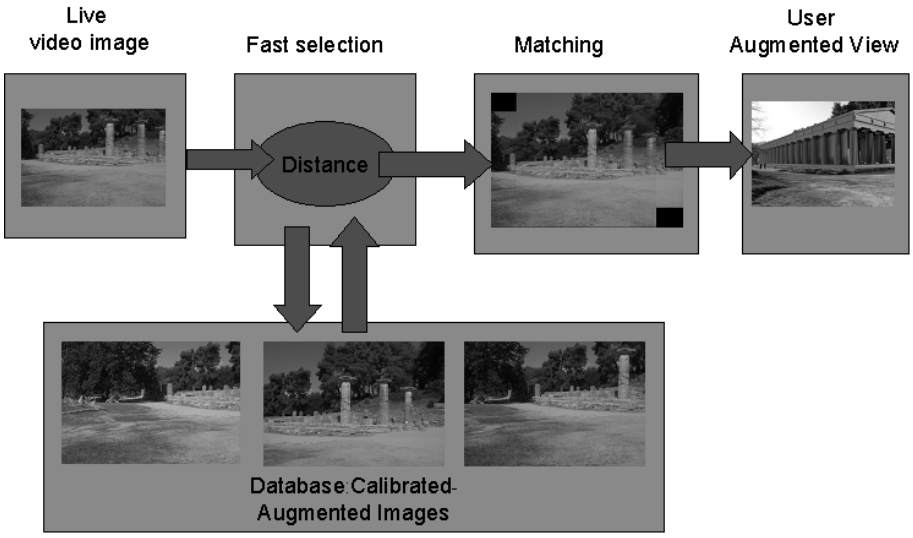



Abbildung 6.46: Übersicht der neuen Trackingarchitektur

Die am Anfang des Kapitels vorgeschlagene Architektur wird leicht modifiziert, in dem das "Selektionsmodul" eingeführt wird. Eine Übersicht der wichtigsten Tracking-Module ist Abbildung 6.46 gegeben.

### 6.5.2 Fourierbasierte Bildselektion

Die Selektion des passenden Referenzbildes muss selbstverständlich unabhängig von dessen Rotation, Skalierung und Verschiebung sein. Im Abschnitt 6.4 wurde gezeigt, dass die Log-Polar-Amplitudenspektren der Bilder unabhängig von der Translation sind und dass sich Rotation und Skalierung auf einen Verschiebungsvektor reduzieren lassen, siehe Gleichung 6.20.
An dieser Stelle werden zur Datenreduzierung und Erstellung einer Rotation und Skalierung unabhängige Bildsignatur-Methoden, die der Methode des intensitätsbasierten Registrierungsverfahren ähneln, siehe Abschnitt6.3, angewendet.
Die Projektion des Log-Polar-Amplitudenspektrums $M(s, \phi)$ auf die zwei Achsen $(s, \phi)$ erzeugt zwei 1D-Signaturen $S(s)$ und $S(\phi)$, die jeweils von $\phi$ beziehungsweise $s$ unabhängig sind. Dabei stellt $s$ den Skalierungsfaktor dar und wird durch die Gleichung $s = \log(r)$ beschrieben. Die Berechnung der Signaturen $S(s)$ und $S(\phi)$ erfolgt wie folgt:

$$S(s) = \sum_{\phi} M(s, \phi) \tag{6.22}$$

und

$$S(\phi) = \sum_{s} M(s, \phi) \tag{6.23}$$

Ein wesentlicher Vorteil dieser Vorgehensweise ist, dass beide Signaturen auf Grund der Datenreduktion aus nur wenig Koeffizienten bestehen. Wie bei dem intensitätsbasierten Verfahren kann damit der Bildvergleich stark beschleunigt werden.
Nach der Berechnung von $S(s)$ und $S(\phi)$ muss der Ähnlichkeitswert für die zwei gegebenen Bilder $f_i$ und $f_j$ bestimmt werden. Dafür wird die euklidische Distanz zwischen den Vek-

toren $S_i(s)$ und $S_j(s)$ einerseits und zwischen $S_i(\phi)$ und $S_j(\phi)$ andererseits berechnet. Der endgültige Ähnlichkeitswert $D_{ij}$ wird gewonnen, indem der Mittelwert beider Distanzen gebildet wird. Mathematisch beschrieben ist $D_{ij}$ wie folgt definiert:

$$D_{ij} = \frac{1}{2}(\sqrt{\sum_{s}(S_i(s) - S_j(s))^2} + \sqrt{\sum_{\phi}(S_i(\phi) - S_j(\phi))^2}) \tag{6.24}$$

Mit dieser Methode kann jedes Live-Videobild mit allen Referenzbildern verglichen und jeweils der Ähnlichkeitswert $D$ berechnet werden. Als Referenzbild für die Bildregistrierung wird das Bild mit dem kleinsten Koeffizient $D$ ausgewählt.

### 6.5.3 Ergebnisse

Die fourierbasierte Pre-Selektionsmethode wird anhand Abbildung 6.47 veranschaulicht und getestet.

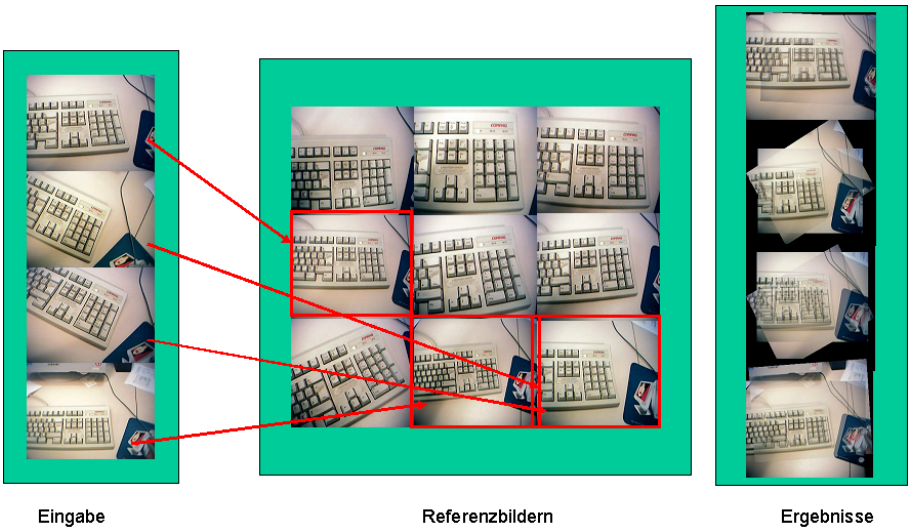



Abbildung 6.47: Testbeispiel des FFT-Selektionsmoduls

Die Referenzbilder bestehen aus neun Ansichten einer Computer-Tastatur. Die linke Spalte enthält vier Bilder, die als Datenquelle fungieren. In der Mitte sind alle Referenzbilder dargestellt. Die Ergebnisse der Selektion und der Registrierung sind in der rechten Spalte abgebildet.
Trotz der starken Drehungen, die in den Quellbildern vorhanden sind, wurden mit Hilfe des beschriebenen Registrierungsverfahrens die Bilder entsprechend der optimalen Überlagerung ausgewählt und erfolgreich registriert.

## 6.6 Anwendungen und Evaluierungen

### 6.6.1 Anwendungsszenario I: Grab and Edit

Das erste Anwendungstool ermöglicht, aus einem Live-Videostrom ein Referenzbild festzuhalten, 2D-Informationen, wie beispielsweise Text oder Overlays auf dem Bild zu editieren

und das Ergebnis zu speichern. Das Tracking kann anschließend eingeschaltet werden und die Informationen auf die Live-Bilder übertragen werden. Somit können beispielsweise Dokumente oder Notizen im Raum oder an bestimmte Objekte angehängt werden. Das System läuft mit einem Framerate von 20 Hz auf einem PC mit 550 MHz CPU.

### 6.6.2 Anwendungsszenario II: AR für den mobilen Einsatz und Außenanwendungen

#### Das ArcheoGuide Projekt

Das Projekt *ArcheoGuide* (Augmented Reality based Cultural Heritage On-site GUIDE) wurde von der europäischen Union finanziert (Projekt IST-1999-11306) und wurde für den Zeitraum von Januar 2000 bis Oktober 2002 bewilligt. Neben dem Fraunhofer-IGD sind das Zentrum für Graphische Datenverarbeitung (Darmstadt), CCG (Portugal), die Firma Intracom (Griechenland), Post Reality (Griechenland), A&C 2000 (Italien) und das griechische Kulturministerium beteiligt.

(a) (b) (c)

Abbildung 6.48: Test des ArcheoGuide-Prototypes

Das ArcheoGuide-System stellt ein mobiles, multimediales Informationssystem dar, das dem Besucher archäologischer Stätten neue Wege der Wissensvermittlung eröffnet. Durch ein mobiles Endgerät werden ihm multimediale Informationen (Bild, Text und Ton) ortsabhängig bereitgestellt. Zusätzlich werden mit Hilfe von Augmented-Reality-Technologien virtuelle Monumente in eine Datenbrille lagerichtig in der Umgebung eingeblendet. Der Besucher blickt so beispielsweise auf eine Ruine, während der Computer ihm die virtuell rekonstruierten Gebäude in ihrer ursprünglichen Pracht präsentiert.
Die mobile Rechnereinheit stellt die zentrale Komponente des Systems dar. Sie wird in einer Schultertasche getragen und beinhaltet zusätzlich zu einem kleinen Rechner ein Head-Mounted-Display (HMD) mit Kamera und ein GPS-System, das als Navigationshilfe dient, siehe Abbildung 6.48(a). Das HMD übernimmt die Funktionalität eines Fernglases, in dem 3D-virtuelle Informationen eingeblendet werden, siehe Abbildung 6.48(b).
Mit dieser Ausrüstung kann der Besucher an bestimmten, vordefinierten “AR- Stationen” z.B. virtuelle Gebäude oder 3D-Animationen von Sportkämpfen, wie in Abbildung 6.49 dargestellt, betrachten.
Da an historischen Stätten in der Regel keine besonderen Sendestationen angebracht oder optische Veränderungen durch Anbringen zusätzlicher Markierung durchgeführt werden dürfen, wurde dieses Problem mit der entwickelten “Tracking mit Referenzbilder”-Methode gelöst.


(a) (b) (c)

Abbildung 6.49: Drei View-Points von (a) Philipeon (b) Hera Tempel und (c) Stadium mit einem Diskuswerfer

Das Tracking wurde auf dem Gelände vom Olympia getestet und evaluiert [4, 27, 24, 89, 25]. Als Rechnerplattform wurde in diesem Fall ein Laptop (Toshiba, 800 Mhz) verwendet. Die Videobilder wurden von einer USB-Kamera aufgenommen, wobei eine Auflösung von $320 \times 240$ Pixel bei einer Framerate von 15 Hz gewählt wurde.
Die entwickelte Applikation läuft komplett in Echtzeit mit der genannten Framerate. Die Augmentierungen bestehen aus 2D-Gif-Bildern und Gif-Animationen, die für die jeweiligen AR-Stationen mit hoher Auflösung vorgeneriert wurden. Abbildung 6.48 zeigt beispielsweise eine Aufnahme vom HMD des Benutzer vor Ort, bei der ein virtueller Diskuswerfer neben einem Tourist eingeblendet wird. Eine komplette Videosequenz, die die Robustheit des Trackings über längere Zeit demonstriert, wird durch Abbildung 6.50 präsentiert.
Das ArcheoGuide-System wurde mit dem Grand-Prix des Jury und der Prix für "Learning/Education" vom Laval-Virtual Festival 2002 prämiert.

### 6.6.3 Anwendungseinschränkungen des markerlosen Echtzeit-Trackings

Auf Grund des fourierbasierten Verfahrens bleibt die Bildregistrierung auf eine 2D-euklidische Transformation beschränkt. Damit können nur spezielle Kamerabewegungen, die hauptsächlich aus einer Kamerarotation resultieren, erfasst werden. Solche Kamerabewegungen liegen beispielsweise bei Verwendung einer Kamera auf einem Stativ oder einer HMD-Anwendung mit bekannter Benutzerposition vor. In diesem Fall ist ohne die Stützung von Markern möglich, die Benutzerbewegungen zu verfolgen und die Umgebung mit zusätzlichen virtuellen Objekten zu erweitern.
Eine Schwierigkeit für das System stellen zur Zeit noch starke Beleuchtungsunterschiede zwischen dem Referenzbild und dem aktuellen Videobild dar. Durch die Veränderung des Sonnenstandes und der Beleuchtungsintensität werden im Bezug auf das Referenzbild andere Schatten erzeugt, die die Bildregistrierung stören können.
Die im Projekt ArcheoGuide durchgeführten Evaluierungen haben dennoch gezeigt, dass sehr gute Ergebnisse erzielt werden konnten. Besondere Bildstrukturen, wie beispielsweise die Tempelsäulen des Hera-Tempels, in Abbildung 6.44 präsentiert, stellten eine starke Struktur dar, wodurch die Registrierung stabilisiert wird. Schnelle rückartige Kamerabewegungen stellen auf Grund der Stützung mit Referenzbildern kein Problem mehr dar. In der Applikation konnte der Benutzer seine Position problemlos um ca. einen Meter nach vorne bzw. nach hinten verlagern, siehe Abbildung 6.49(b). Die Bewegungen wurden durch den Skalierungsfaktor abgefangen. Als für das Verfahren kritisch stellten sich jedoch


Abbildung 6.50: Live Videosequenz vom View-Point "Philipeon-Tempel"

seitliche Bewegungen, die den stabilen Bereich eines halben Meters überschritten, heraus.

## 6.7 Zusammenfassung

In diesem Kapitel wurde eine Lösung des markerlosen optischen Trackings präsentiert. Insbesondere wurde zuerst die Notwendigkeit einer sogenannten *Stützung* für optische Trackingsysteme dargelegt und anschließend das Konzept des Trackings auf Basis von Referenzbildern abgeleitet. Den Kern dieser neuen Technologie stellt die Bildregistrierung dar. Nach Analyse der möglichen Ansätze wurden zwei Vorgehensweisen ausgewählt und im Detail entsprechend der Zielsetzung evaluiert. Die fourierbasierte Lösung lieferte die besseren Ergebnisse und ermöglichte eine praktische Umsetzung des vorgestellten Trackingkonzeptes. Ein Modul zur schnellen Bild-Selektion (image retrieval) wurde entwickelt und implementiert und damit das Tracking auf Basis mehrerer Referenzbilder erweitert. Das gesamte Trackingsystem wurde für Mobil- und Outdoor-AR-Anwendungen im Rahmen des ArcheoGuide-Projektes eingesetzt und ist in der Lage, Live-Videos mit einer Bildrate von 20 Hz um virtuelle Objekte zu erweitern.

# Kapitel 7

# Zusammenfassung und Ausblick

## 7.1 Zusammenfassung

Im Rahmen dieser Arbeit wurden Computer-Vision-basierte Kalibirerungs- und Trackingverfahren für Augmented-Reality-Anwendungen entwickelt und evaluiert.
Für Offline-Anwendungen wurden insbesondere neue Methoden zur Berechnung der Kameraposition und -orientierung auf Basis von 3D-Punkten umgesetzt und anhand ausführlicher Simulationen verglichen. Die besten Algorithmen wurden in das Tool *CamCal*, einem Tool zur Kalibrierung und Erweiterung einzelner Bilder, integriert. Anschließend wurden auf Basis von Verfahren aus dem Bereich *structure and motion* neue Methoden zur Erweiterung von Bildsequenzen entwickelt. Diese Methoden besitzen den Vorteil, dass auf 3D-Punktdaten verzichten werden kann und die benötigten Informationen allein aus den Bildern extrahiert werden können. Mit Hilfe der implemtierten Algorithmen können sowohl planare als auch nicht-planare Szenen behandelt werden. Darüber hinaus wurde ein neues Verfahren mit dem Name *Calibration Propagation* konzipiert und umgesetzt. Die Zielsetzung dieses Verfahrens besteht darin, die intrinsischen Kameraparameter eines unbekannten Bildes auf Basis eines zweiten Bildes, für welche alle Kameraparameter bekannt sind, zu kalibrieren und mit virtuellen Objekten zu erweitern. Das Verfahren wurde *Calibration Propagation* genannt, da die Kalibrierungsdaten von einem kalibrierten auf ein unkalibriertes Bild übertragen werden. Durch dieses Verfahren können in einer Bildsequenz sowohl Position und Orientierung als auch der Zoomfaktor einer Kamera geändert werden, ohne dass weitere Informationen als die Bilder für die Rückgewinnung aller erforderlichen Parameter benötigt werden.
Für die Bearbeitung kompletter Videosequenzen wurde ein Automatisierungsmechanismus zur Bestimmung der 2D-Punktposition über allen Bildern entwickelt. Das Verfahren basiert auf eine interaktive Initialisierungsphase, in der eine 3D-Rekonstruktion der zu verfolgenden Punkte vorgenommen wird. Anschließend werden in einer Trackingschleife die 3D- rekonstruierten Punkten rekursiv in die Bilder der Videosequenz reprojiziert und ihre 2D-Position mit Hilfe einer Subpixel- Korrelationsmethode bestimmt. Der 3D-Kamerapfad wird von Bild zu Bild berechnet und anschließend mit einem globalen Optimierungsalgorithmus, auch *Bundle Adjustment* genannt, über die ganze Videosequenz verfeinert. Eine robuste Methode, die auch sogenannte *outliers* zulässt, konnte auf Basis von M-Estimatoren erzielt werden.
Im Bezug auf optisches Echtzeit-Tracking wurde ein marker-basierter Ansatz verfolgt.

Ein neues Verfahren, das auf einem *Punkt-zur-Linie-* Distanzminimierungsverfahren basiert [23], konnte sowohl die Genauigkeitsanforderung erfüllen als auch gute Echtzeit-Performance vorweisen. Weitere Tracking-Methoden wurden im Rahmen des VBT-Systems entwickelt. Das VBT-System basiert auf der Verwendung von sowohl schwarz-weißen als auch farbigen Markern und kann auf einem Laptop mit herkömmlichen Kameras betrieben werden.
Da bei Out-Door-Anwendungen die Entfernungen zur Szene und den dazugehörigen Objekten zu groß sind, kann bei diesen Anwendungen nicht mehr auf Marker zur Trackingunterstützung zurückgegriffen werden. Für dieses Problem wurde ein markerloses Trackingverfahren für mobile und Out-Door-AR-Systeme entworfen und entwickelt [28, 22, 27, 24, 89, 25]. Im Rahmen der Arbeit wurde zuerst der Begriff der Stützung für optische Tracker eingeführt und definiert. Die Stützung wird üblicherweise von Markern gegeben und wurde für das markerlose Tracking durch Bilder der Umgebung zur Verfügung gestellt. Diese Bilder stellen die Referenzbilder des Trackings dar und, werden kalibriert und anschließend mit virtuellen Informationen erweitert. Während der Laufzeit wird das Live-Videobild mit allen gespeicherten Bildern verglichen, das Bestpassende selektiert und anschließend die Transformation vom Referenzbild zum Live-Video-Bild mit Hilfe von Bildregistrierungsverfahren berechnet. Das Registierungsverfahren wurde im Rahmen dieser Arbeit intensiv erforscht, wobei Ansätze auf Intensitäts- und Fourierbasis ausgewählt und implementiert wurden. Das Tracking konnte mit dem Fourier-Ansatz erfolgreich umgesetzt und demonstriert werden.

## 7.2 Ausblick

Als Ausblick werden in diesem Abschnitt verschiedene Themen für weitere mögliche Forschungsarbeiten vorgestellt. Das erste Forschungsgebiet beinhaltet das bildbasierte Online-Tracking ohne Marker. In diesem Bereich besteht Forschungsbedarf, um die Verfahren zur vollständigen Bestimmung der 3D-Kameraposition und -orientierung zu optimieren. Eine Möglichkeit besteht darin, nicht mehr Bilder der Umgebung, sondern ein 3D-texturiertes Modell als Stützung für das Tracking zu verwenden und aus diesem Modell alle Positions- und Orientierungsparameter der Kamera zu ermitteln.
Weiterer Forschungsbedarf existiert auch auf dem Gebiet des hybriden Trackings. In Kombination mit den vorgestellten optischen Verfahren können weitere Sensoren angewendet und dadurch das Tracking weiter verbessert werden. Durch Inertial-Sensoren können beispielsweise Positions- und Orientierungswerte auch bei Verdeckung eines Großteils der Szene an das AR-System weitergegeben werden.
Abschließend sei als mögliches Forschungsgebiet das Thema *selbstlernende Trackingverfahren* genannt. Durch die Weiterentwicklung des Systems in diesem Bereich könnte das Trackingsystem autonom in einer Initialisierungsphase oder während der Laufzeit die Daten erfassen, die für die Erweiterung der Bilder benötigt werden. Durch diese Technik können Flexibilität und Nutzbarkeit des Trackingsystems vergrößert und so auch, neue Anwendungsgebiete für AR eröffnet werden.

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Literaturverzeichnis


[1] R. Jain A. Gupta. Visual information retrieval. *Communication of ACM*, May 1997.

[2] M.A. Abidi and T. Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation. *PAMI*, 17(5):534–538, May 1995.

[3] Adnan Ansar and Daniilidis Konstantinos. Linear pose estimation from points or lines. In *ECCV'02*, pages 282–296, 2002.

[4] ArcheoGuide. Archeoguide, the augmented reality based cultural heritage on-site guide. BMBF, http://www.archeoguide.gr, 2002.

[5] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. *PAMI*, 9(5):698–700, September 1987.

[6] Ron Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, Nov/Dec 2001.

[7] R.T. Azuma. A survey of augmented reality. *Presence, Special Issue on Augmented Reality*, 6(4):355–385, August 1997.

[8] Atkison K. B. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, 1996.

[9] Ramesh Raskar Bandyopadhyay Deepak and Henry Fuchs. Dynamic shader lamps: Painting on real objects. In *IEEE International Symposium on Augmented Reality, ISAR '01*, New York, US, 2001.

[10] D.K. Bhatnagar. Position trackers for head mounted display systems: survey. Technical Report TR93-010, Department of Computer Science, University of North Carolina - Chapel Hill, March 1 1993. Wed, 26 Jun 1996 18:27:25 GMT.

[11] Pascal Brand. *Reconstruction tridimensionnelle a partir d´une camera en mouvement:influence de la precision*. PhD thesis, LIFIA-IMAG-INRIA-Rhone-Alpes, Oktober 1995.

[12] I.N. Bronstein and K.A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Zuerich, Frankfurt, Thun, 1976.

[13] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.

[14] R. Mohr C. Schmid and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[15] Rodrigo L. Carceroni and Christopher M. Brown. Numerical methods for model-based pose recovery. Technical Report TR659, University of Rochester, Computer Science Department, August 1997. Thu, 21 Aug 97 19:58:23 GMT.

[16] David Casasent and Demetri Psaltis. Position-, rotation-, and scale-invariant optical correlation. *Applied Optics*, 15(7):1795–1799, July 1976. Mellin Transform, Fourier Transform, Fourier-Mellin Transform.

[17] D.W. Caudell, T.P.; Mizell. Augmented reality: an application of heads-up display technology to manual manufacturing processes. *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, 2:659 –669, 1992.

[18] Youngkwan Cho, Jun Park, and Ulrich Neumann. Fast color fiducial detection and dynamic workspace extension in video see-through self-tracking augmented reality. *Proceedings of the Fifth Pacific Conference on Computer Graphics and Applications*, 1997.

[19] Curtis D., Mizell D., Gruenbaum P., and Janin. Several devils in the details: Making an ar app work in the airplane factory, 1998.

[20] Dementhon. Model based object pose in 25 lines of code. *International Journal on Computer Vision*, 3, 1995.

[21] M. Dhome, M. Richetin, J.T. Lapreste, and G. Rives. Determination of the attitude of 3-d objects from a single perspective view. *T-PAMI*, 11:1265–1278, 1989.

[22] Stricker Didier. Tracking with reference images: A real-time and markerless tracking solution for out-door augmented reality applications. In *Virtual Reality, Archaeology, and Cultural Heritage, VAST'01*, Nov. 2001.

[23] Stricker Didier, Klinker Gudrun, and Reiners Dirk. A fast and robust line-based optical tracker for augmented reality applications. In *First International Workshop on Augmented Reality*. Springer Verlag, 1998.

[24] Stricker Didier, Karigiannis John, Christou T. Ioannis, Gleue Tim, and Ioannidis Nikos. Augmented reality for visitors of cultural heritage sites. In *CAST*, 2001.

[25] Stricker Didier, Karigiannis John, Vlahakis Vassilios, Dähne Patrick, and Ioannidis Nikos. Archeoguide - a mobile augmented reality system for archeological sites - a solution to the tracking problematic. In *Electronic Imaging and Visual Arts, EVA*, Nov. 2002.

[26] Stricker Didier and Navab Navab. Calibration propagation for image augmentation. In *Proceedings of Second International Workshop on Augmented Reality*, San Francisco, 1999.

[27] Stricker Didier, Dähne Patrick, Seibert Frank, Christou Ioannis, Almeida Luis, Carlucci Renzo, and Ioannidis Nicos. Design and development issues for archeoguide: An

augmented reality based cultural heritage on-site guide. In *International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging*. ACM, 2001.

[28] Stricker Didier and Kettenbach Thomas. Out-door augmented reality: From scene preparation to markerless vision based tracking. In *International Symposium on Augmented Reality*. IEEE-Press, 2001.

[29] Stricker Didier, Fröhlich Torsten, and Söllner-Eckert Claudia. The augmented man. In *International Symposium on Augmented Reality*. IEEE, 2000.

[30] Reiners Dirk, Stricker Didier, Klinker Gudrun, and Mueller Stefan. Augmented reality for construction tasks: Doorlock assembly. In *First International Workshop on Augmented Reality*. Springer Verlag, 1998.

[31] O.D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[32] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. MIT Press, 2001.

[33] S. Feiner, B. MacIntyre, T. Hollere, and A. Webster. A touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment, 1997.

[34] Paul D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001.

[35] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[36] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P.Yanker. Query by image and video content: The qbic system. *IEEE Computer*, pages 23–32, September 1995.

[37] Golub G. and Van Loan C. *Matrix Computations*. The Johns Hopkins University Press, 1983.

[38] R. Gonzalez and P. Wintz. Digital image processing addisonwesley publish company, 1987.

[39] Klinker Gudrun, Stricker Didier, and Reiners Dirk. The use of 3d model in augmented reality. In Springer Verlag, editor, *3D Structure from Multiple Images of Large-Scale Environments, European Workshop at ECCV'98 SMILE 98*. R. Koch L. Van Gool, 1998.

[40] Klinker Gudrun, Stricker Didier, and Reiners Dirk. Augmented reality: A balance act between high quality and real-time constraints. In *First International Symposium on Mixed Reality (ISMR'99)*. Y. Ohta and H. Tamura (eds.), in Mixed Reality - Merging Real and Virtual Worlds, March 1999.

[41] Klinker Gudrun, Stricker Didier, and Reiners Dirk. An optically based direct manipulation interface for human-computer interaction in an augmented world. In Michael Gervaut, Dieter Schmalstieg, and Axel Hildebrand, editors, *Virtual Environments '99. Proceedings of the Eurographics Workshop in Vienna, Austria*, pages 53–62. Springer-Verlag Wien, 1999.

[42] Klinker Gudrun, Stricker Didier, and Reiners Dirk. Augmented reality for exterior construction applications. In W. Barfield and T. Caudell, editors, *Augmented Reality and Wearable Computers*. Lawrence Erlbaum Press, 2000.

[43] R. M. Haralick. Determining camera parameters from the perspective projection of a rectangle. *Pattern Recognition*, 22:223–230, 1989.

[44] R.M. Haralick, H. Joo, C.N. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *SMC*, 19(6):1426–1446, November 1989.

[45] R.M. Haralick, C.N. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the 3-point perspective pose estimation problem. *IJCV*, 13(3):331–356, December 1994.

[46] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[47] R.I. Hartley. In defense of the eight-point algorithm. *PAMI*, 19(6):580–593, June 1997.

[48] O.D. Huang, T.S.; Faugeras. Some properties of the e matrix in two-view motion estimation. *PAMI*, 11(12):1310 –1312, December 1989.

[49] Heikkilä J. and Silven O. A four-step camera calibration procedure with implicit image correction. In *Proc. CVPR*, pages 1106–1112. IEEE, 1997.

[50] Maybank S. J. *3D Reconstruction*. Springer Verlag, 1992.

[51] M. Jethwa, A. Zisserman, and A. Fitzgibbon. Real-time Panoramic Mosaics and Augmented Reality . Ninth British Machine Vision Conference, 1998.

[52] K. Kanatani. Geometric computation for machine vision. In *Oxford University Press*, 1993.

[53] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. Automated camera calibration and 3d egomotion estimation for augmented reality applications. In *Proc. CAIP '97*, Kiel, Germany, September 1997.

[54] Zhong-Dan Lan. *Methodes robustes en vision: application aux appariements visuels*. PhD thesis, 1997.

[55] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs N.J., 1974.

[56] H. Li, B.S. Manjunath, and S.K. Mitra. A contour-based approach to multisensor image registration. *IP*, 4(3):320–334, March 1995.

[57] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[58] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:441–450, 1991.

[59] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.

[60] Q. T. Luong and T. Vieville. Canonic representations for the geometries of project views. Technical Report CSD-93-772, University of California, Berkeley, 1993.

[61] D.W. Marquardt. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters.* J.Soc. Indust. Appl. Math, 11(2):431-441, 1963.

[62] P. Meer, D. Mintz, D.Y. Kim, and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, April 1991.

[63] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D(12), December 1994.

[64] D.W. Mizell. Virtual reality and augmented reality in aircraft design and manufacturing. *WESCON/94. Idea/Microelectronics. Conference Record*, page 91, 1994.

[65] Matthias Muhlich and Rudolf Mester. The role of total least squares in motion analysis. In *ECCV (2)*, pages 305–321, 1998.

[66] OpenSG. Opensg, an open source scene graph. http://www.opensg.org, 2000.

[67] N. Otsu. A threshold selection method from grey-level histograms. *SMC*, 9(1):62–66, January 1979.

[68] T. Phong, R. Horaud, A. Yassine, and P. Tao. Object pose from 2-d to 3-d point and line correspondences. *International Journal of Computer Vision*, 15:225–243, 1995.

[69] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, 2 edition, 1992.

[70] Long Quan and Zhong-Dan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, August 1999.

[71] Luong Quang-Tuan and Faugeras Olivier. The fundamental matrix: theory, algorithms, and stability analysis. *IJCV*, 1994.

[72] K. P. Spinnler R. W. Frischholz. A class of algorithms for real-time subpixel registration. In *Europto Conference*, Munich.

[73] B.S. Reddy and B.N. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IP*, 5(8):1266–1271, August 1996.

[74] J. Rekimoto and K. Nagao. The world through the computer: Computer augmented interaction with real world environments. In *Proc. UIST '95*, pages 29–36, 1995.

[75] Azuma Ron and Bishop Gary. Improving static and dynamic registration in an optical see-through HMD. In *Proc. Siggraph '94*, pages 197–204, Orlando, FL, July 1994.

[76] E. Rose, D. Breen, K.H. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, and D. Greer. Annotating real-world objects using augmented reality. In *Proc. Computer Graphics: Developments in Virtual Environments*. Academic Press Ltd, 1995.

[77] B. Rousso, S. Peleg, and I. Finci. Mosaicing with Generalized Strips. ARPA Image Understanding Workshop, pp. 255-260, 1997.

[78] Micel Cherbuliez Ruggero Milanese and Thierry Pun. Invariant content-based image retrieval using the fourier-mellin transform. 1999.

[79] M.R. Shortis, Clarke T. A., and Short T. A. A comparison of some techniques for the subpixel location of discrete target images. In *Videometrics III*, pages 239–250, Boston, MA, 1994. SPIE.

[80] H. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment, 1998.

[81] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R.W. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence, Special Issue on Augmented Reality*, 6(4):386–398, August 1997.

[82] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic mosaics and environment maps. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 251–258. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[83] Jean-Philippe Tarel. Calibration de caméra fondée sur les ellipses. Technical Report RR-2200, INRIA Rocquencourt, 1994.

[84] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. *Proc. International Conference on Computer Vision (ICCV'99)*, July 1999.

[85] R.Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. CVPR*, pages 364–374. IEEE, 1986.

[86] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *PAMI*, 6(1):13–27, January 1984.

[87] D. Tsen and C. Chang. Color segmentation using perceptual attributes. In *11th IAPR Int. Conf. on Pattern Recognition*, pages 228–231, 1992.

[88] Tuceryan and Navab. Single point active alignment method (spaam) for calibrating an optical see-through head mounted display. In *IEEE International Symposium on Augmented Reality, ISAR '00*, Munich Germany, 2000.

[89] Vassilios Vlahakis, Nikolaos Ioannidis, John Karigiannis, Didier Stricker, Patrick Daehne, and Luis Ameida. Archeoguide: Challenges and solutions for a persolanized augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications*, pages 34–47, Sept.-Nov. 2002.

[90] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *PAMI*, 14(10):965–980, 1992.

[91] Z. Zhang. Understanding the relationship between the optimization criteria in two-view motion analysis. *Proc. International Conference on Computer Vision (ICCV'98)*, January 4–7 1998.

[92] Z. Zhang. Flexible camera calibration. *Proc. International Conference on Computer Vision (ICCV'99)*, 1999.

[93] I. Zoghiami, O.P. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *CVPR97*, pages 420–425, 1997.

# Lebenslauf

Didier Stricker
geboren am 23.09.69 in Sarreguemines, Frankreich

| | |
|---|---|
| bis Juli 1987 | Gymnasium, Saverne: Abitur |
| 1987 - 1989 | Universität Metz: Vorbereitungskurs auf die Technische Hochschule (Classes Préparatoires aux Grandes Ecoles) |
| 1989 - 1992 | Elektrotechnikstudium an der Technischen Hochschule: "Ecole Nationale Superieure d'Eléctronique et de Radioéléctricite de Grenoble" |
| 1992 - 1993 | Technische Hochschule Karlsruhe |
| 1993 - 1995 | Wissenschaftlicher Mitarbeiter am Institut für Maschinenwesen im Baubetrieb vom Prof. Dr./Uni.Tokio T. Bock |
| 1995 - 1997 | Wissenschaftlicher Mitarbeiter am Forschungsinstitut für Informationsverarbeitung und Mustererkennung (FIM) in Ettlingen bei Karlsruhe |
| Seit 1997 | Wissenschaftlicher Mitarbeiter am Fraunhofer Insitut für Graphische Datenverarbeitung, Darmstadt |